Object Detection
(Plus some bonuses)

EECS 442 – Jeong Joon Park

Winter 2024, University of Michigan

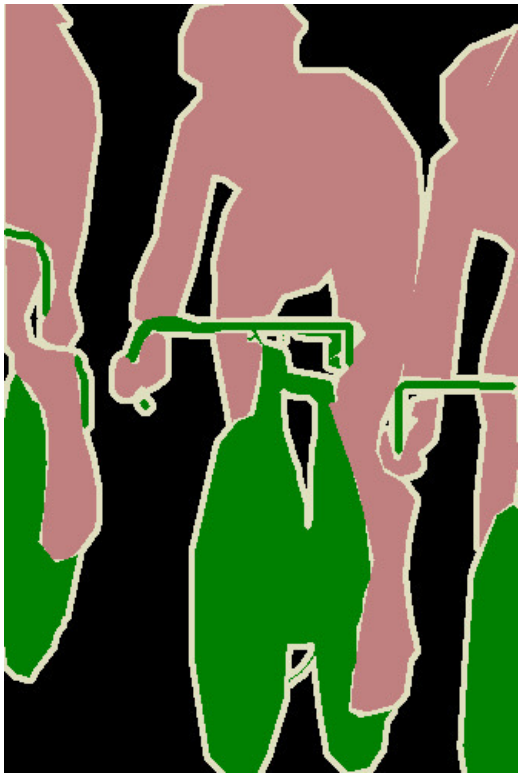
Last Time

"Semantic Segmentation": Label each pixel with the object category it belongs to.

Input

CNN

Target



Today – Object Detection

"Object Detection": Draw a box around each instance of a list of categories

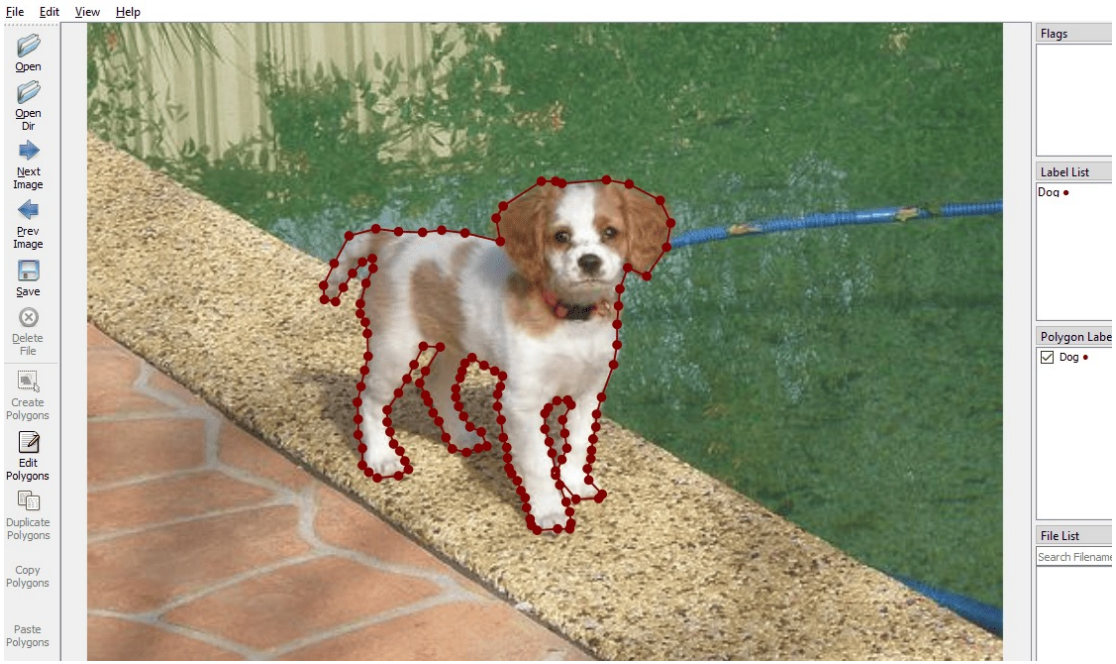
Input

CNN

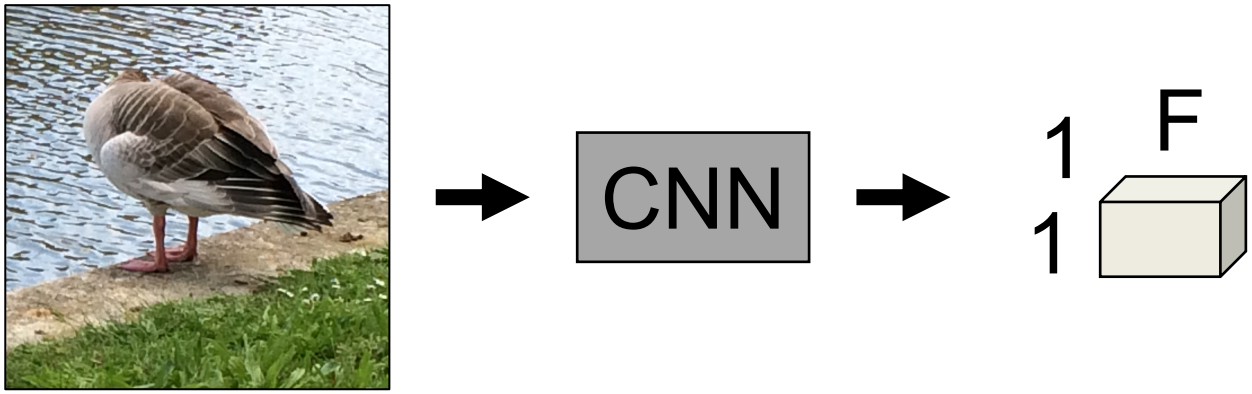
Target

Why Object Detection?

- Instance segmentation?
- Efficiency: real-time applications
- Data annotation requirement



The Wrong Way To Do It

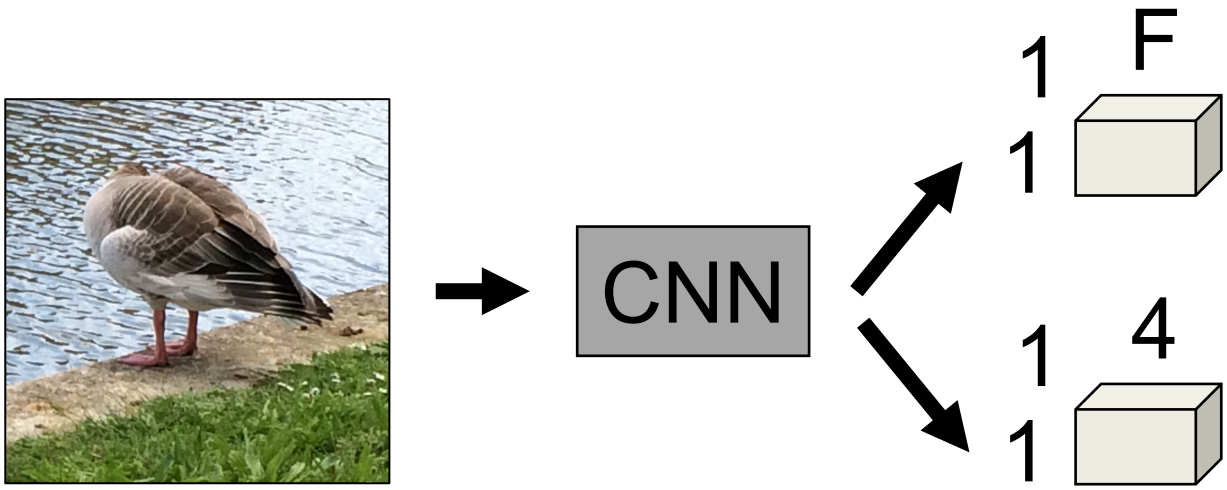


Starting point:

Can predict the probability of F classes

P(cat), P(goose), … P(tractor)

The Wrong Way To Do It

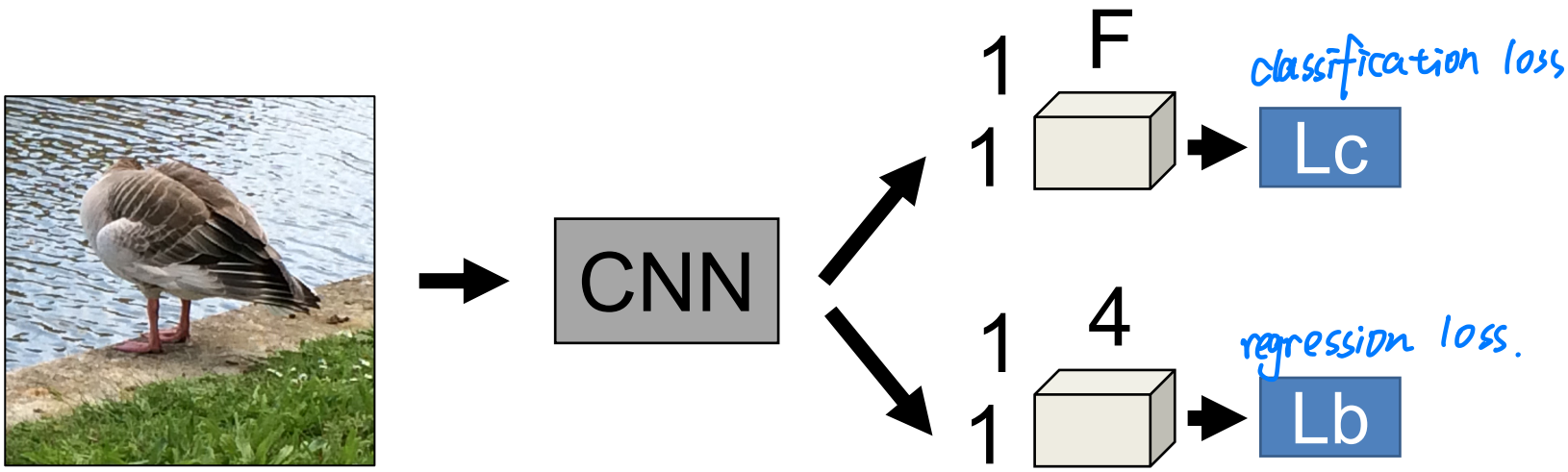


Add another output (why not):

Predict the *bounding box* of the object

[x,y,width,height] or [minX,minY,maxX,maxY]

The Wrong Way To Do It



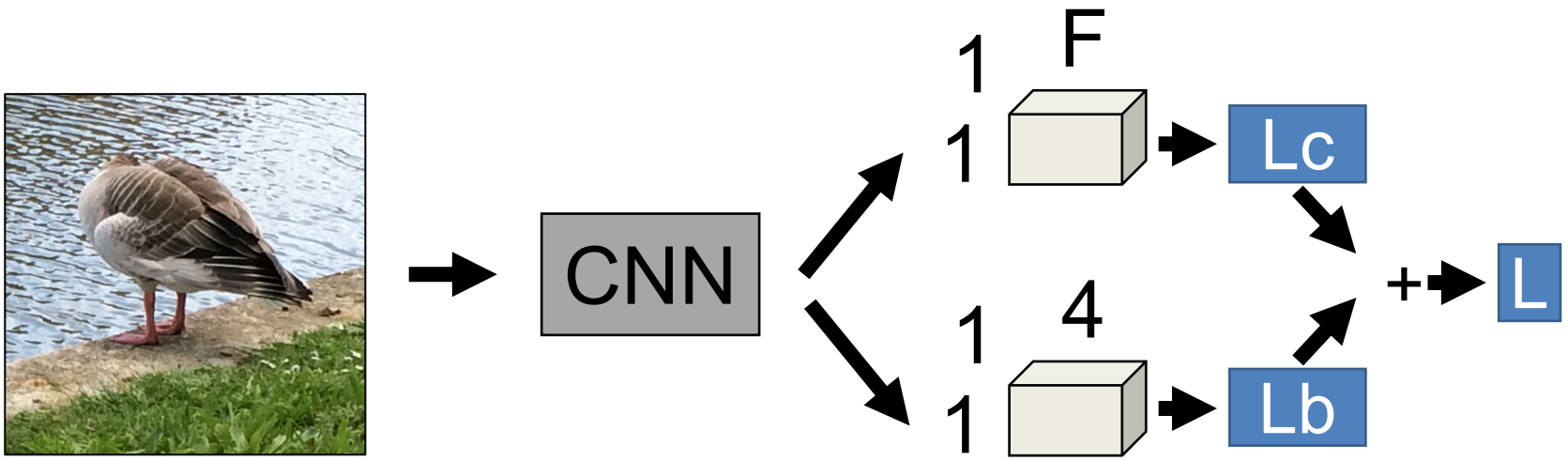
Put a loss on it:

Penalize mistakes on the classes with

Lc = negative log-likelihood

Lb = L2 loss

The Wrong Way To Do It

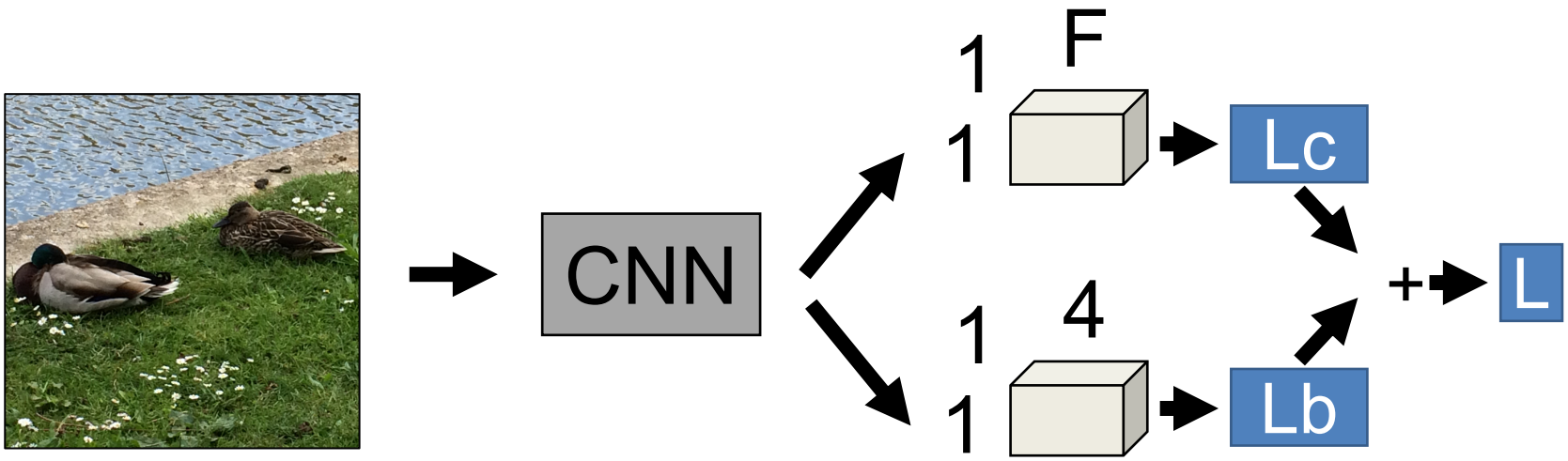


Add losses, backpropagate

Final loss: $L = Lc + \lambda Lb$

Why do we need the λ?

The Wrong Way To Do It

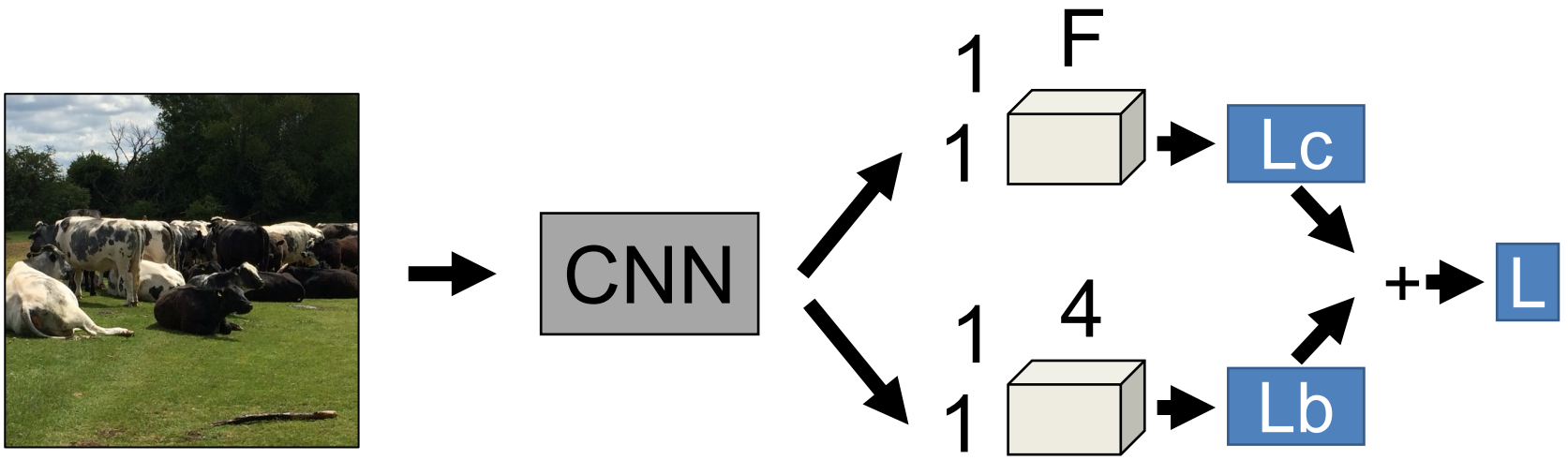


Now there are two ducks.

How many outputs do we need?

F, 4, F, 4 = 2*(F+4)

The Wrong Way To Do It

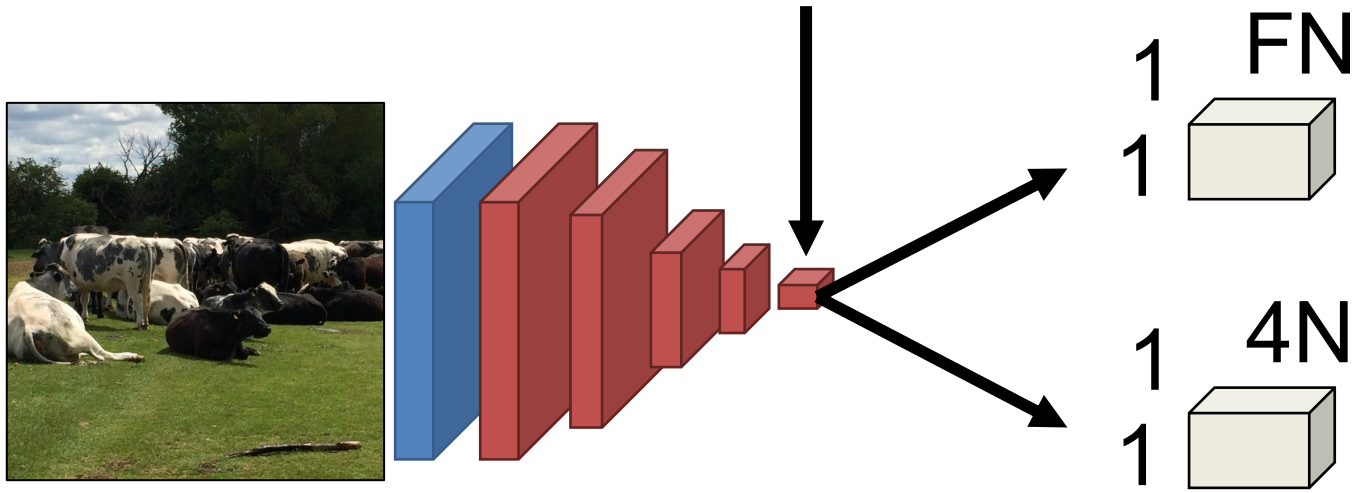


Now it's a herd of cows.
We need *lots* of outputs
(in fact the precise number of objects that are in the image, which is circular reasoning).

In General

- Usually can't do varying-size outputs.
- Even if we could, think about how *you* would solve it if you were a network.

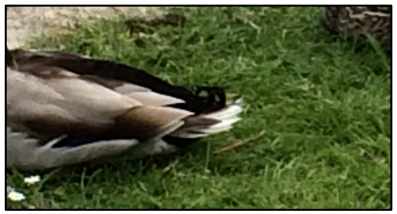
Bottleneck has to *encode* where the objects are for all objects and all N



An Alternate Approach

Examine every sub-window and determine if it is a tight box around an object

Yes



No?

Hold this thought

No

Sliding Window Classification

Let's assume we're looking for pedestrians in a box with a fixed aspect ratio.

Slide credit: J. Hays

Sliding Window

Key idea – just try all the subwindows in the image at all positions.

Slide credit: J. Hays

Generating hypotheses

Key idea – just try all the subwindows in the image at all positions **and scales.**

Note – Template did not change size

Slide credit: J. Hays

Each window classified separately

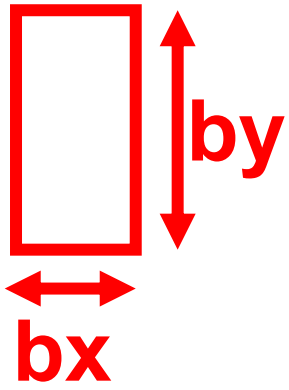
Slide credit: J. Hays

How Many Boxes Are There?

Given a HxW image and a “template” of size by, bx.

Q. How many sub-boxes are there of size (by,bx)?

A. (H-by)*(W-bx)

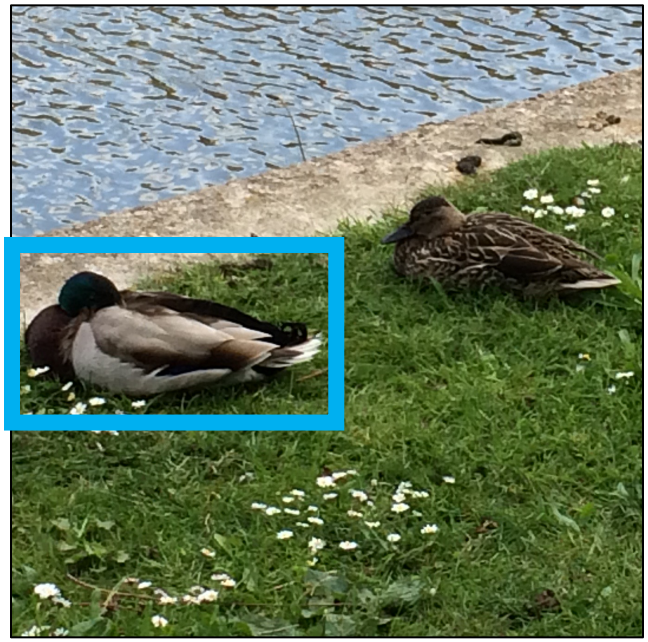


This is before considering adding:

- *scales* (by*s,bx*s)
- *aspect ratios* (by*sy,bx*sx)

Challenges of Object Detection

- Have to evaluate *tons* of boxes
- Positive instances of objects are *extremely* rare

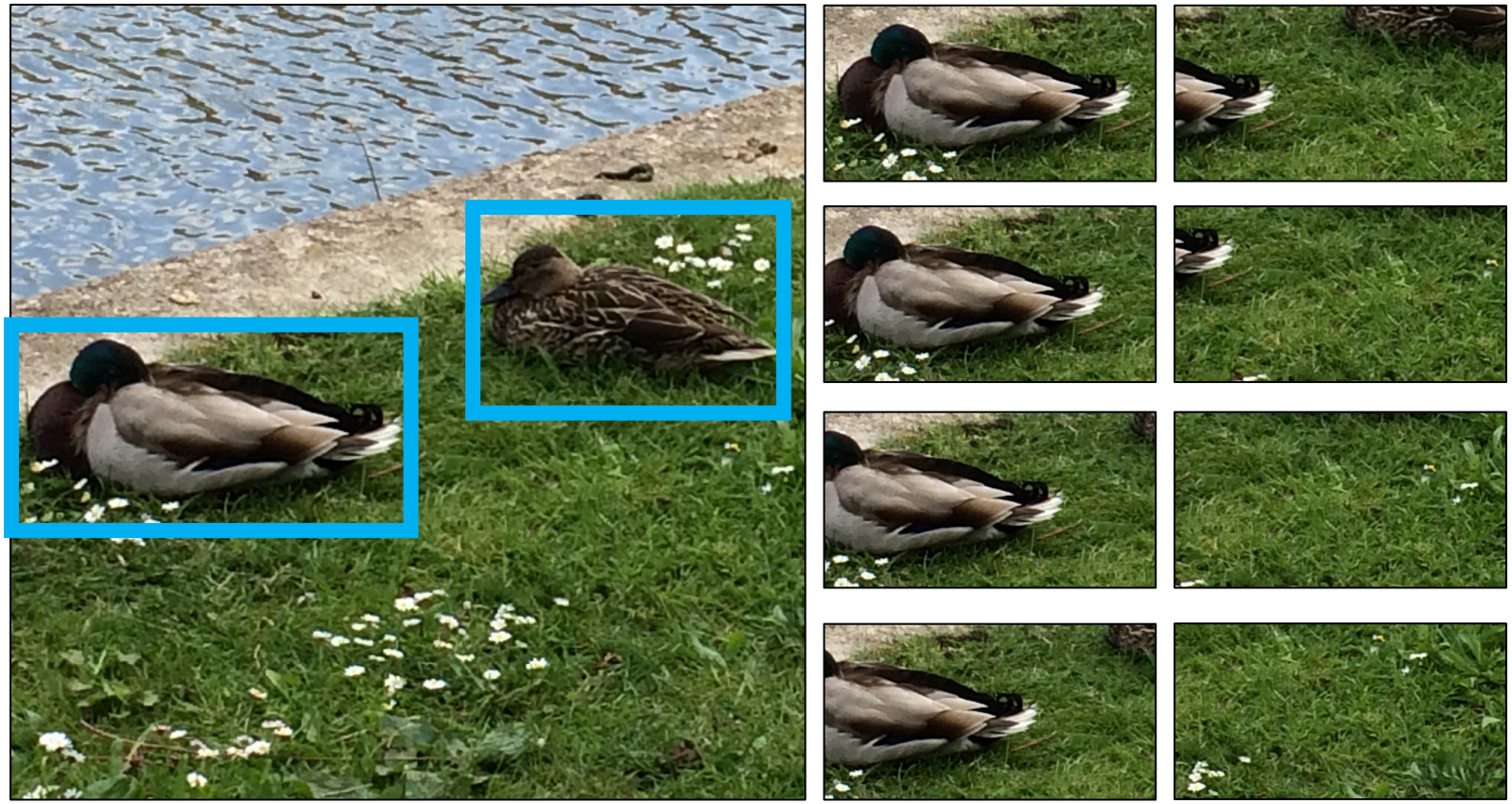


How many ways can we get the box wrong?

1. Wrong left x
2. Wrong right x
3. Wrong top y
4. Wrong bottom y

Evaluating – Bounding Boxes

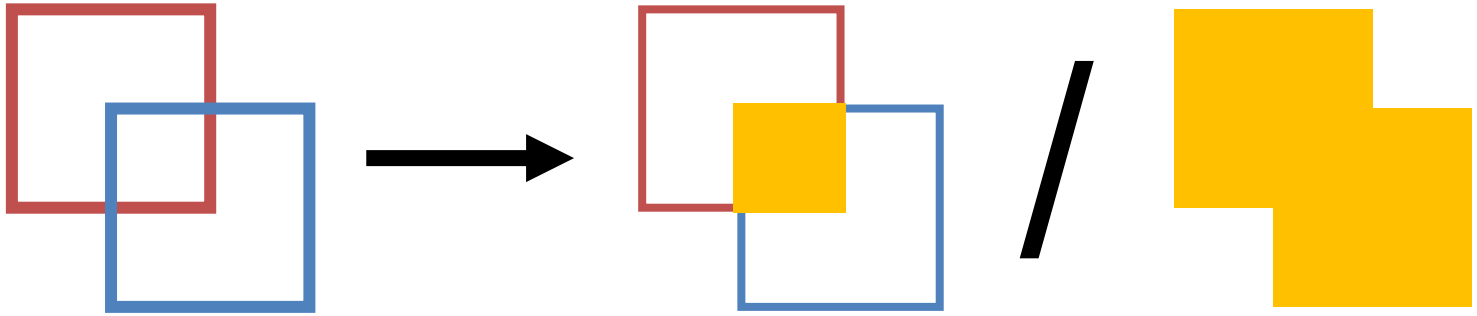
Raise your hand when you think the detection stops being correct.

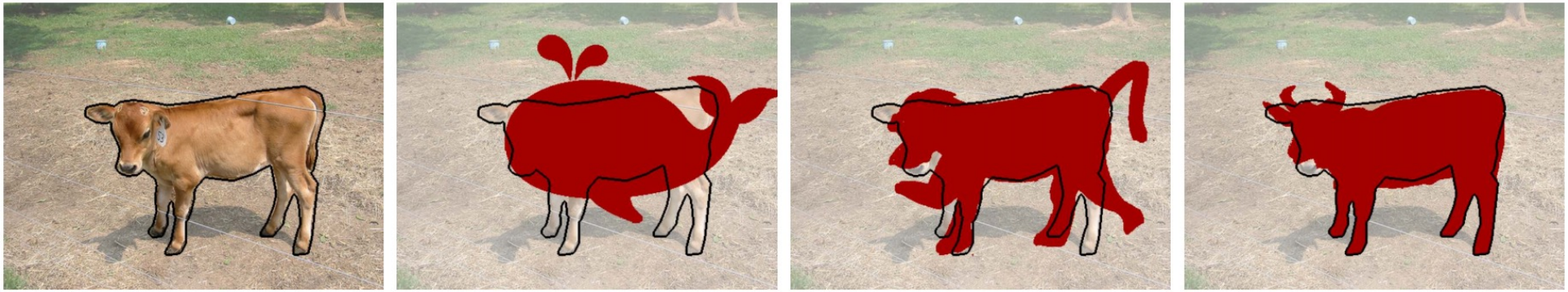


Evaluating – Bounding Boxes

Standard metric for two boxes:

Intersection over union/IoU/Jaccard index





(a) Ground truth (b) $\mathcal{J} = 0.554$ (c) $\mathcal{J} = 0.703$ (d) $\mathcal{J} = 0.910$

Jaccard example credit: P. Kraehenbuehl et al. ECCV 2014

Evaluating Performance

- Remember: accuracy = average of whether prediction is correct
- Suppose I have a system that gets 99% accuracy in person detection.
- **What's wrong?**
- I can get that by just saying no object everywhere!

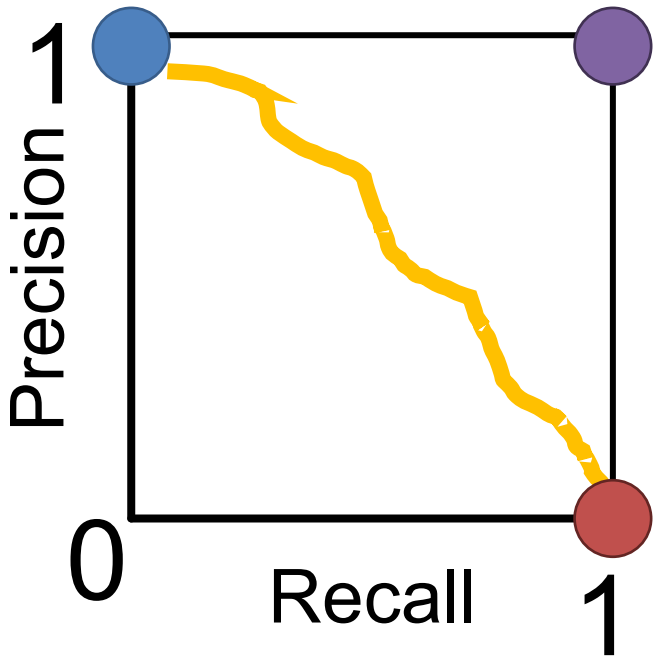
Evaluating Performance

- True detection aka true positive: high IoU (>0.5)
- Precision: #true detections / #detections by detector
- Recall: #true detections / #ground truth positives

if you want to use one number to describe the performance of your detection model.

Reject everything: no mistakes

: Do an integral of this graph under the curve



Ideal!

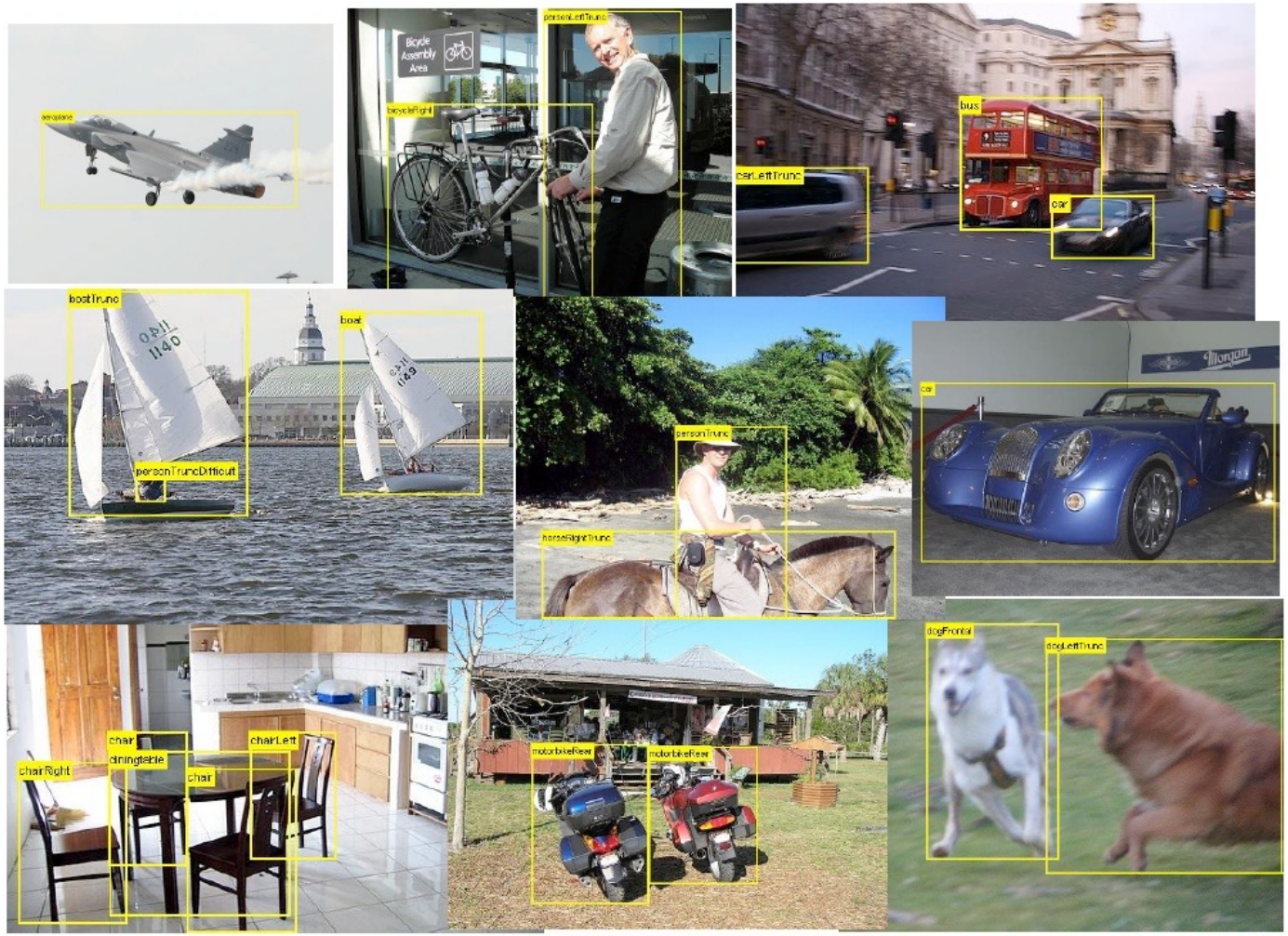
(Averge precision > Ap

if you do "AP" for all the classes ⟶ mean Average precision (MAP).

Summarize by area under curve (avg. precision)

Accept everything: Miss nothing

Generic object detection



Slide Credit: S. Lazebnik

Histograms of oriented gradients (HOG)

Partition image into blocks and compute histogram of gradient orientations in each block

HxWx3 Image

H'xW'xC' Image

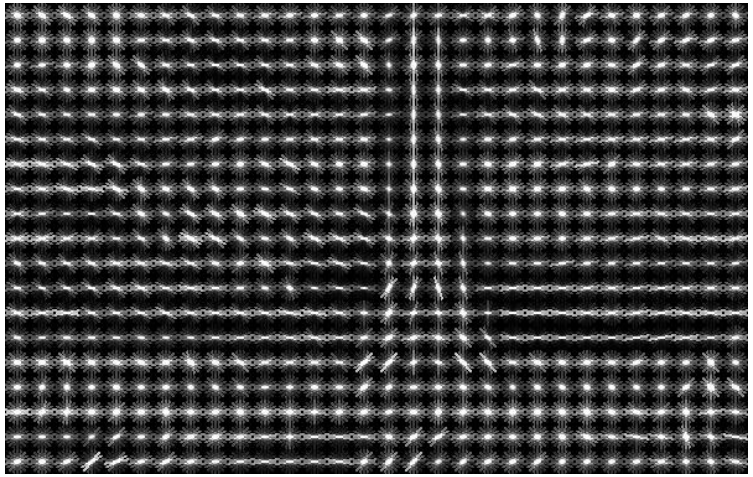


Image credit: N. Snavely

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Slide Credit: S. Lazebnik

Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine

positive training examples

negative training examples

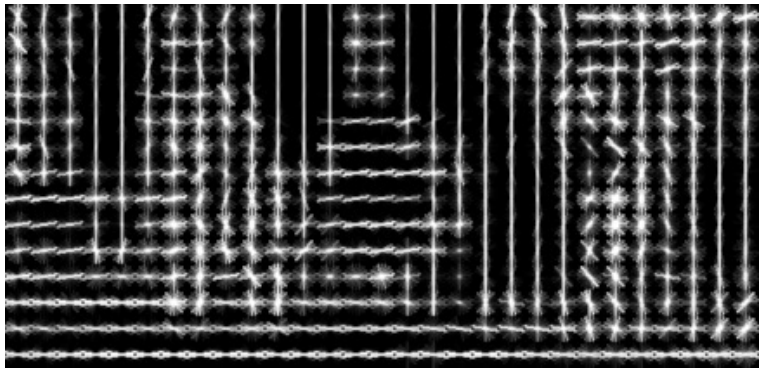
N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Slide Credit: S. Lazebnik

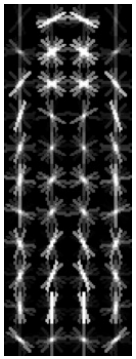
Pedestrian detection with HOG

- Train pedestrian “template” using a linear svm
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG *pyramid*

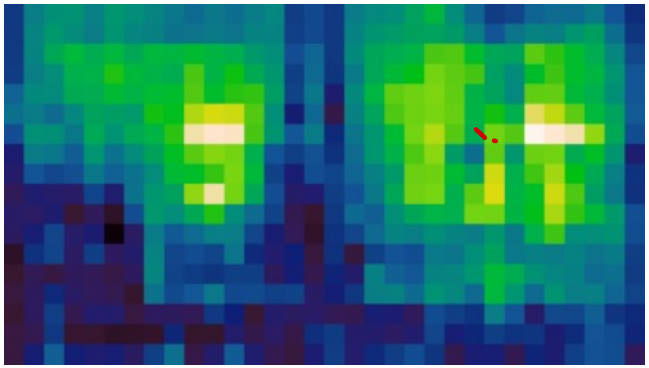
HOG feature map



Template



Detector response map



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Slide Credit: S. Lazebnik

Example detections

[Dalal and Triggs, CVPR 2005]

Slide Credit: S. Lazebnik

PASCAL VOC Challenge (2005-2012)

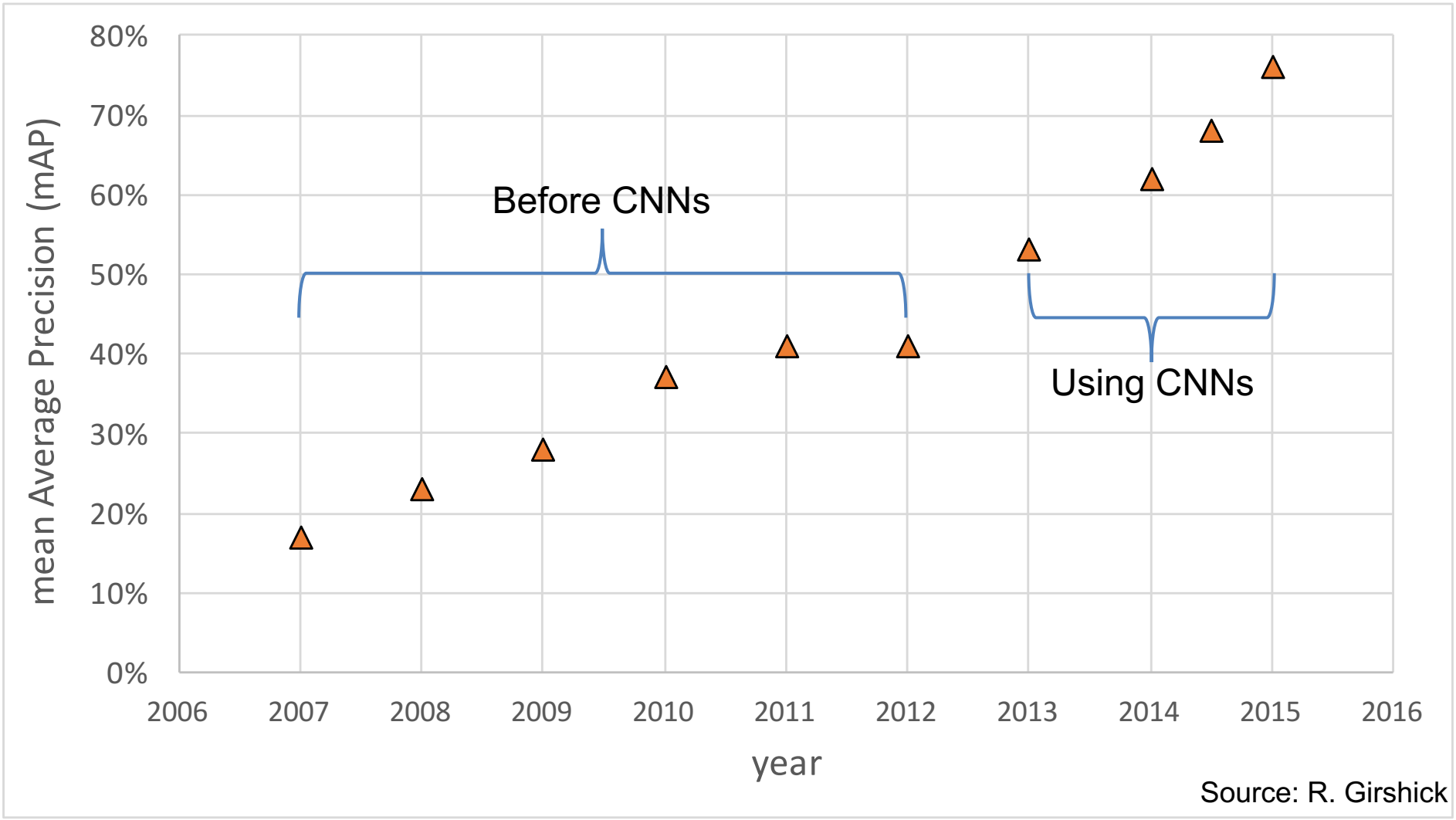
- 20 challenge classes:
- *Person*
- *Animals:* bird, cat, cow, dog, horse, sheep
- *Vehicles:* aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor
- Dataset size (by 2012): 11.5K training/validation images, 27K bounding boxes, 7K segmentations

Slide Credit: S. Lazebnik

http://host.robots.ox.ac.uk/pascal/VOC/

Object detection progress

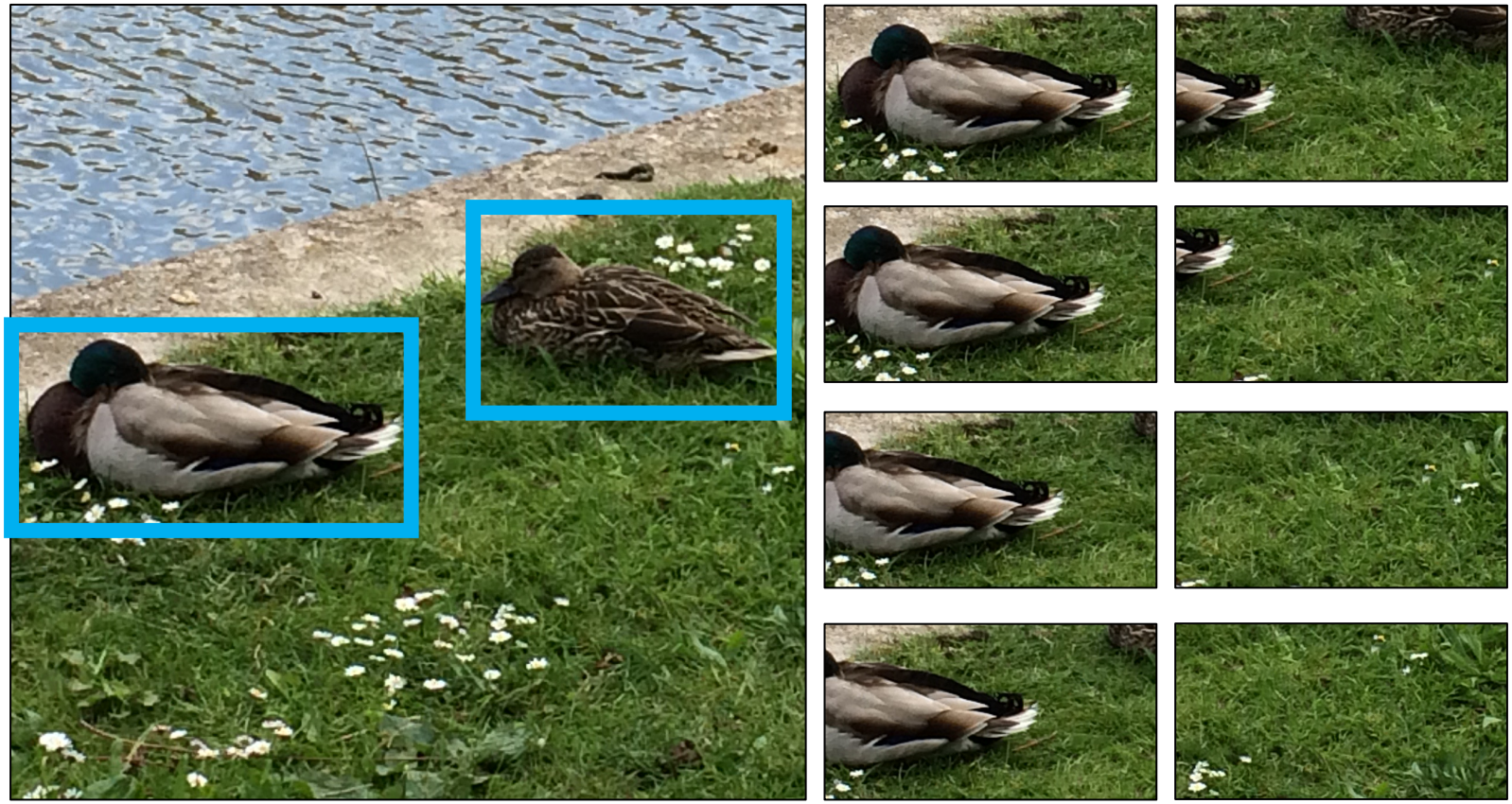
PASCAL VOC



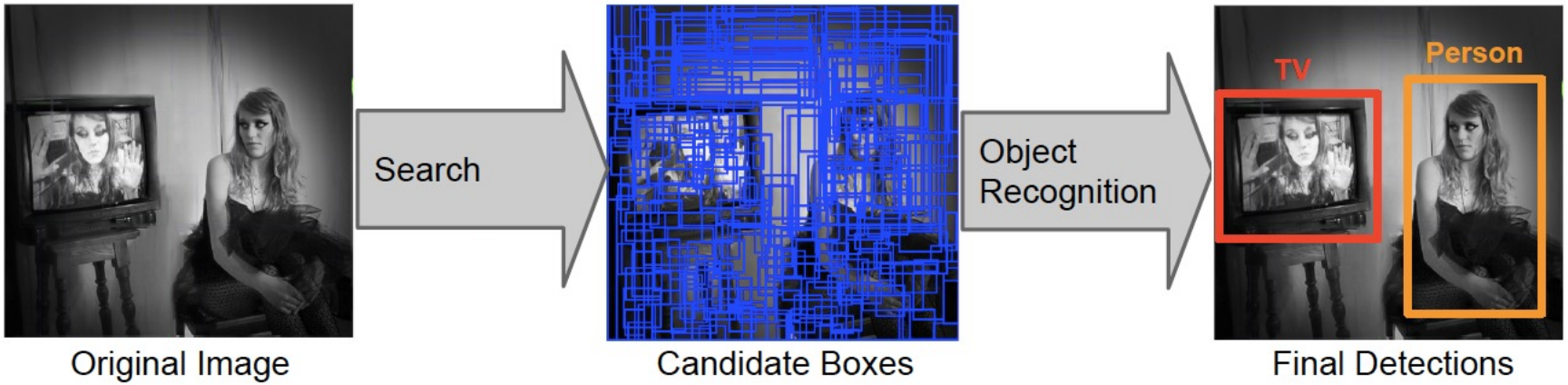
Source: R. Girshick

Region Proposals

Do I need to spend a lot of time filtering all the boxes covering grass?



Region Proposals



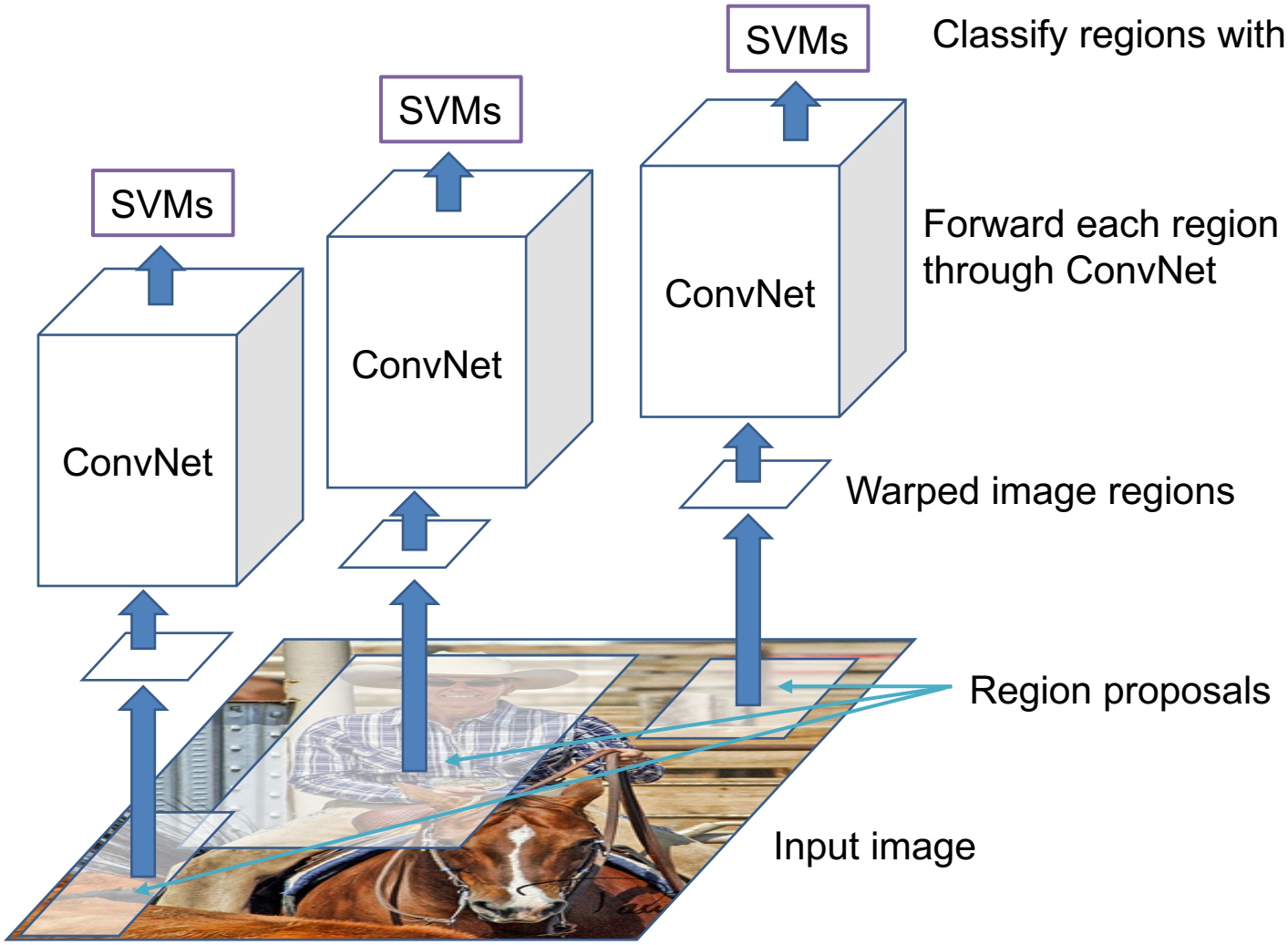
- As an alternative to sliding window search, evaluate a few hundred *region proposals*
  - Can use slower but more powerful features and classifiers
  - Proposal mechanism can be category-independent
  - Proposal mechanism can be trained

Slide Credit: S. Lazebnik

Example Region Proposal Algo: Edge boxes: Locating object proposals from edges. 2014

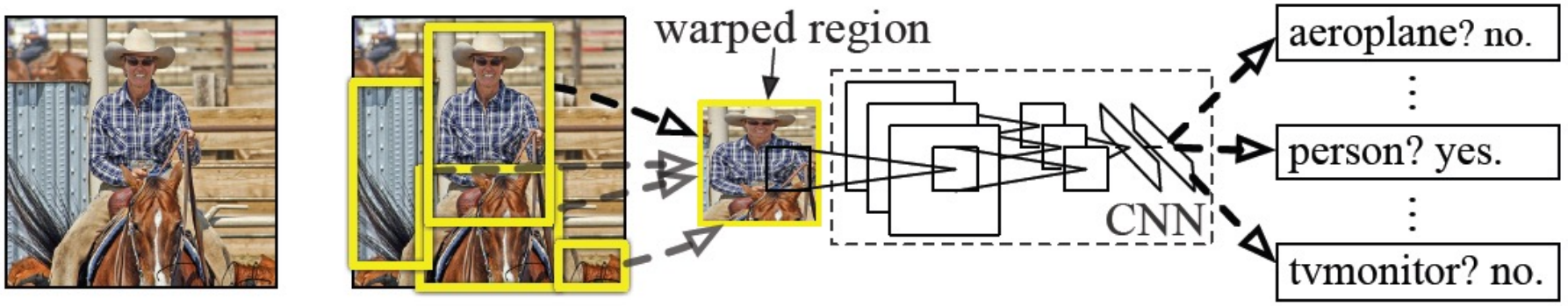
R-CNN: Region proposals + CNN features

Source: R. Girshick



R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.

R-CNN details



- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to refine box locations
- **Performance:** mAP of 53.7% on PASCAL 2010 (vs. 35.1% for Selective Search and 33.4% for DPM).

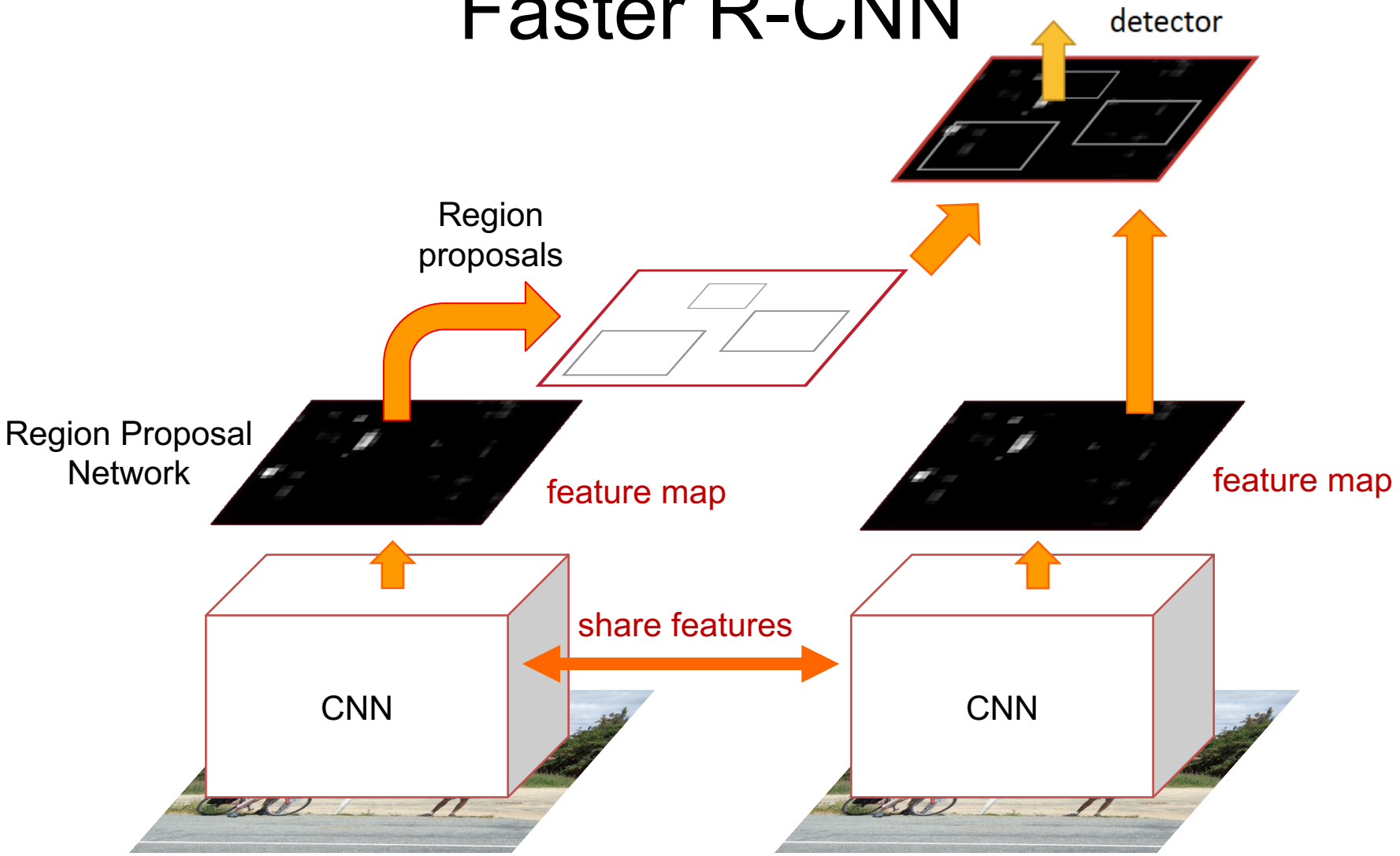
R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.

R-CNN pros and cons

- Pros
  - Accurate!
  - Any deep architecture can immediately be “plugged in”
- Cons
  - Ad hoc training objectives
    - Fine-tune network with softmax classifier (log loss)
    - Train post-hoc linear SVMs (hinge loss)
    - Train post-hoc bounding-box regressions (least squares)
  - Training is slow (84h), takes a lot of disk space
    - 2000 CNN passes per image
  - Inference (detection) is slow (47s / image with VGG16)

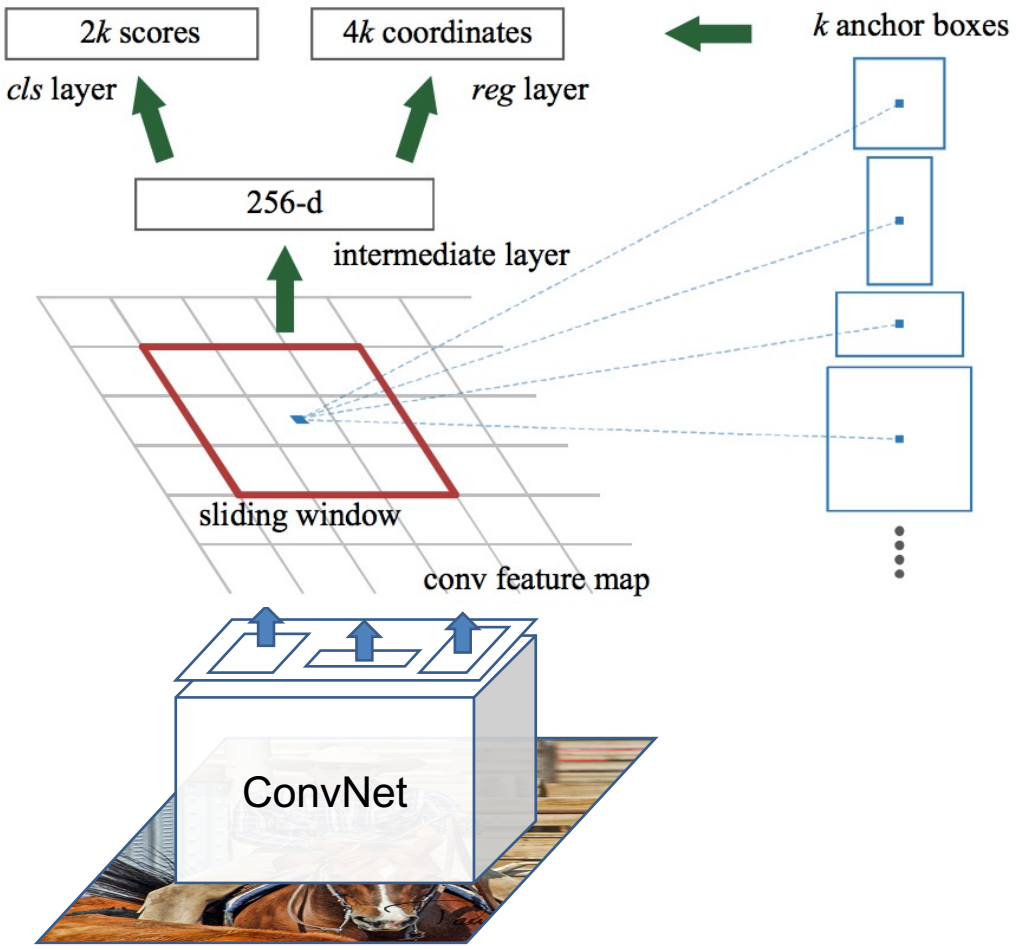
Slide Credit: S. Lazebnik

Faster R-CNN



S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

Region Proposal Network (RPN)



Small network applied to conv5 feature map.

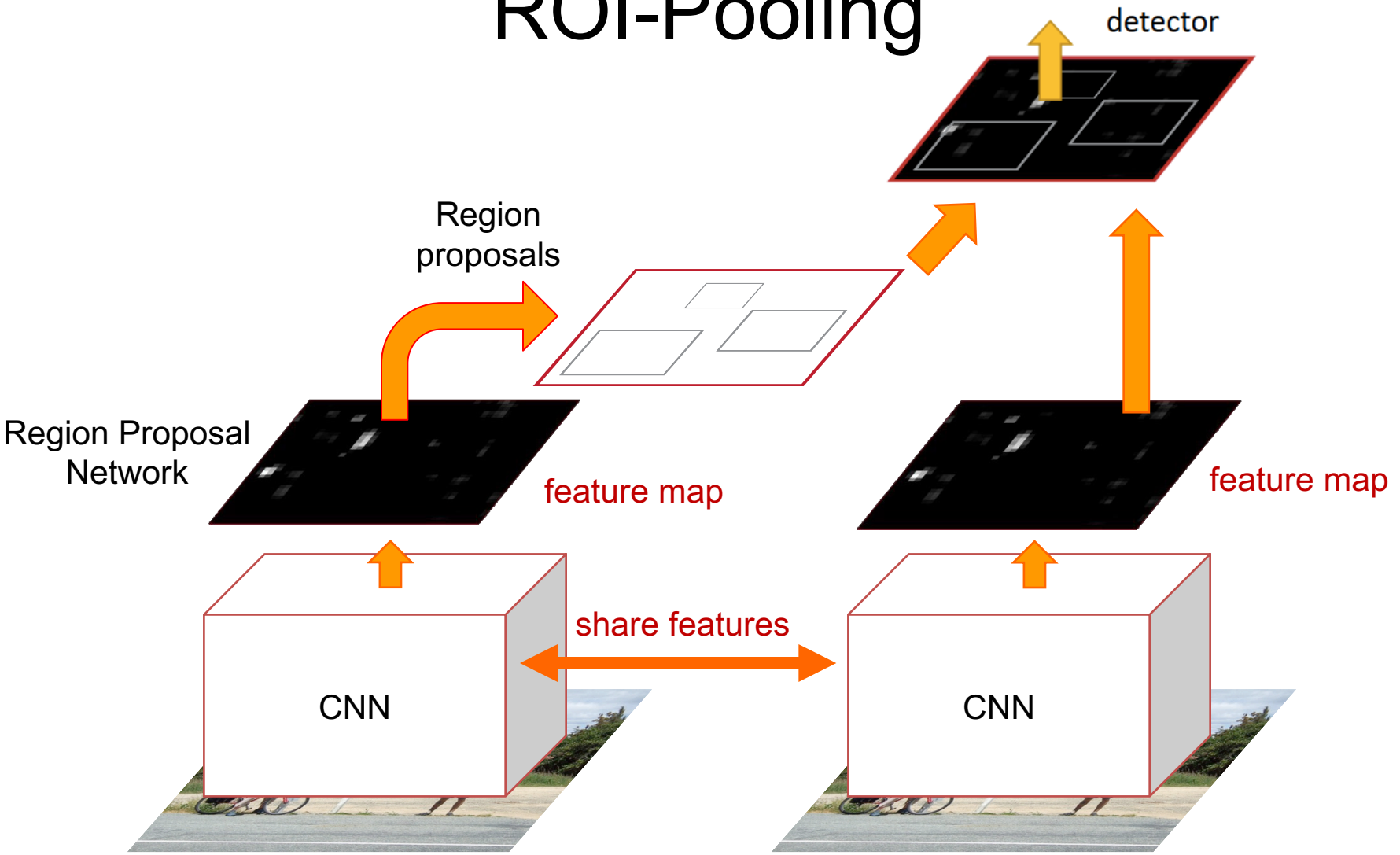
Predicts:

- good box or not (classification),
- how to modify box (regression)

for k “anchors” or boxes relative to the position in feature map.

Source: R. Girshick

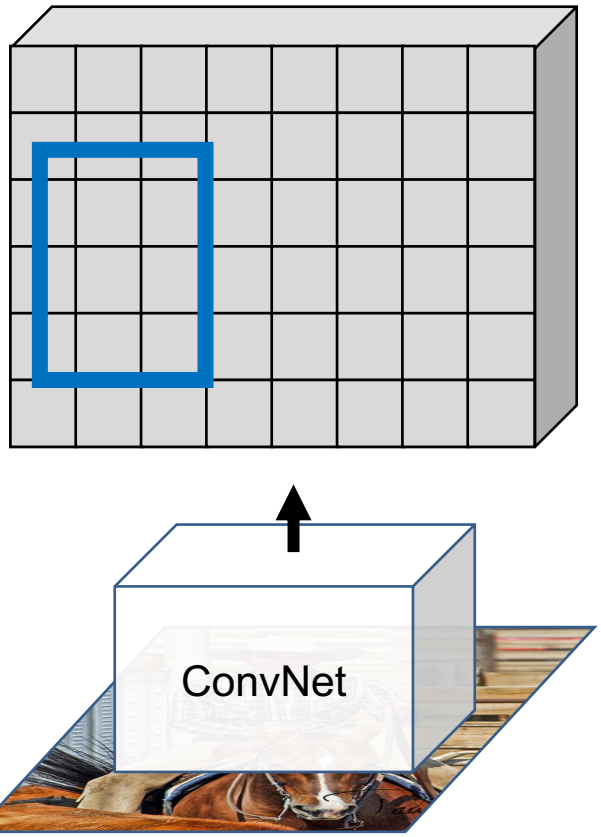
ROI-Pooling



S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

ROI Pooling/Align

Feature Map
(e.g., 6x8x256)

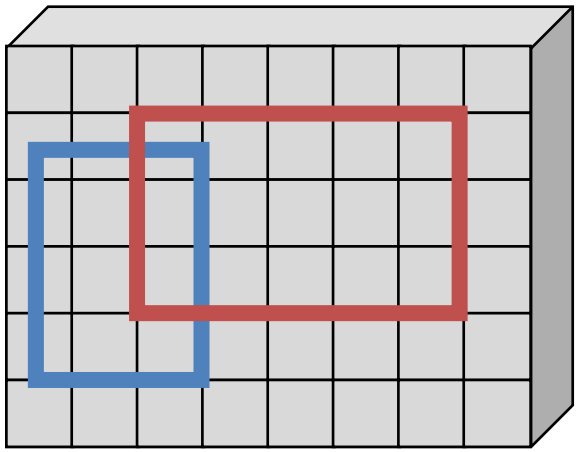


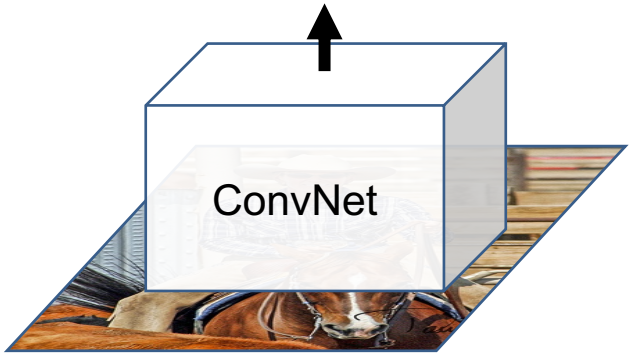
Given box in original image, calculate where the box goes.

- Example: H=600,W=800
- Feature map is H'=6, W'=8
- Box: left x=50, top y=150, width=250, height=350
- Feature map box: left x=0.5, y =1.5, width=2.5, height=3.5

ROI Pooling/Align

Feature Map
(e.g., 6x8x256)

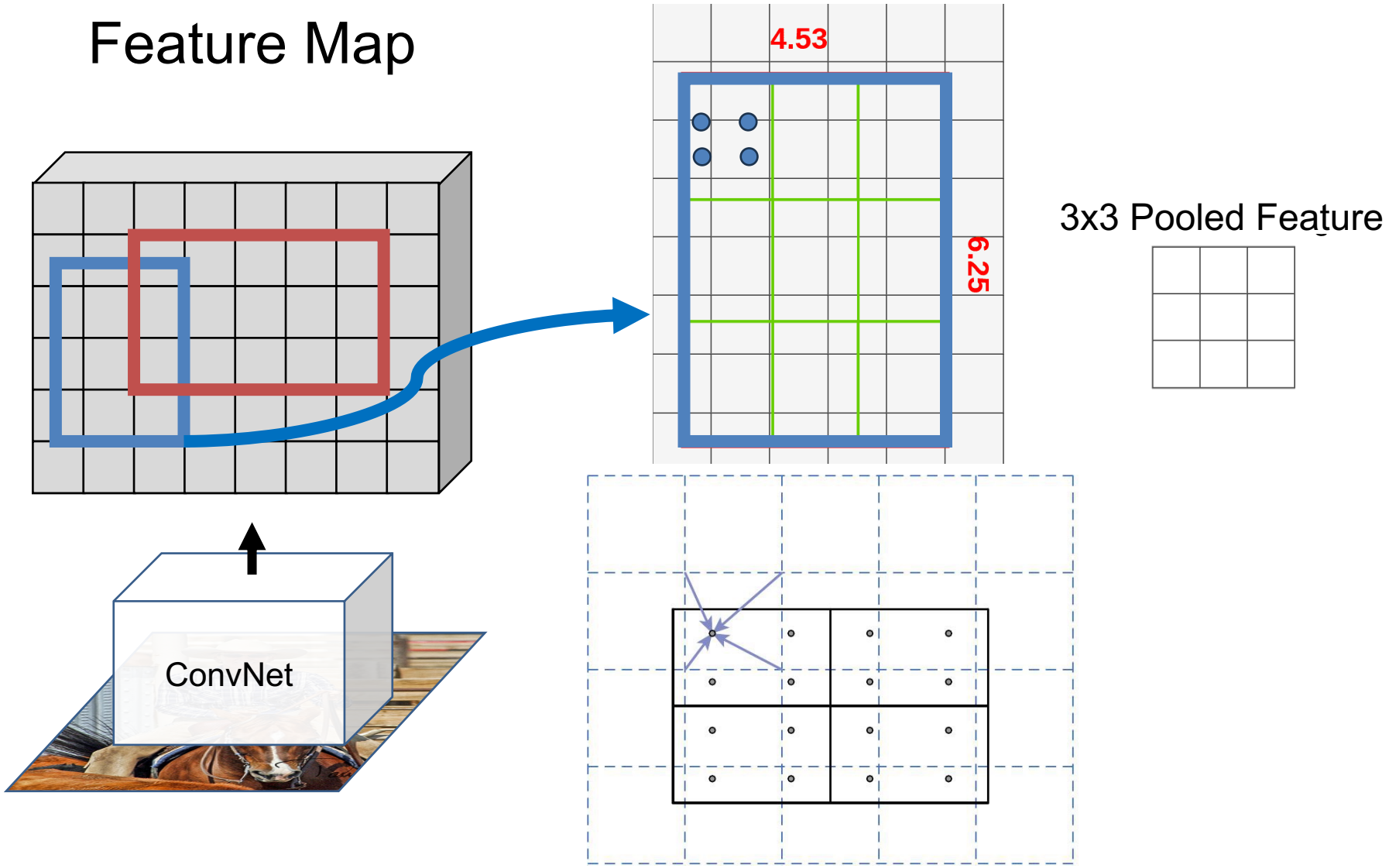




Given box in original image, calculate where the box goes.

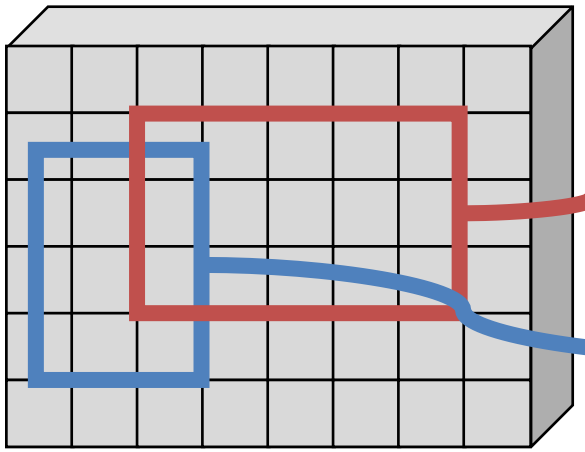
- Example: H=600,W=800
- Feature map is H'=6, W'=8
- Box: left x=50, top y=150, width=250, height=350
- Feature map box: left x=0.5, y =1.5, width=2.5, height=3.5
- Other feature map box: left x=2, top y = 1, width=5, height=3

ROI Pooling/Align

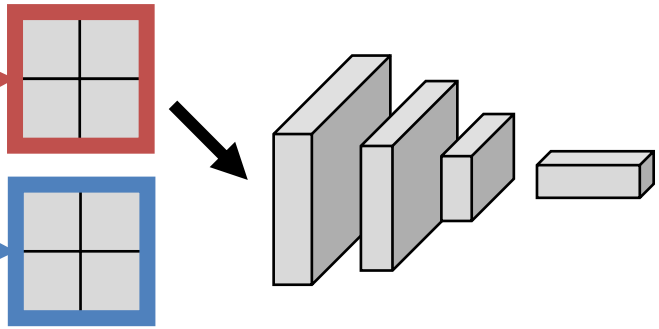


ROI Pooling/Align

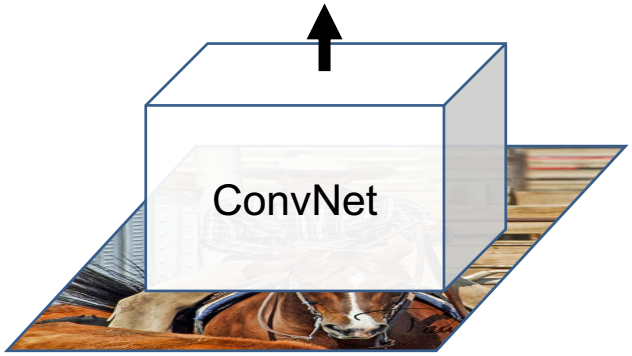
Feature Map
(e.g., 6x8x256)



Resize to fixed size (e.g., 7x7)
Details **critical**, but beyond scope of class.



P(airplane)
P(cat)
…
P(car)
P(zebra)

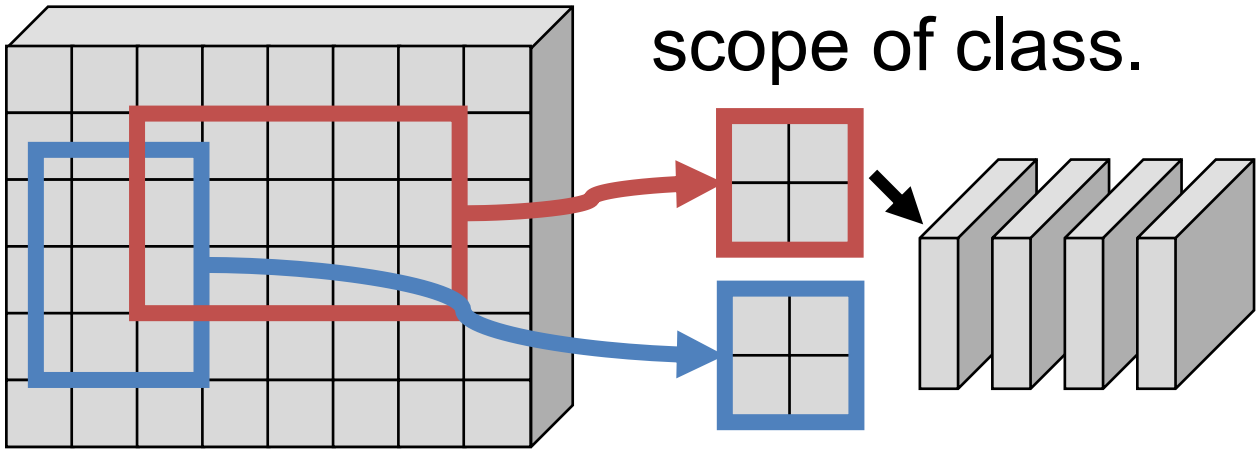


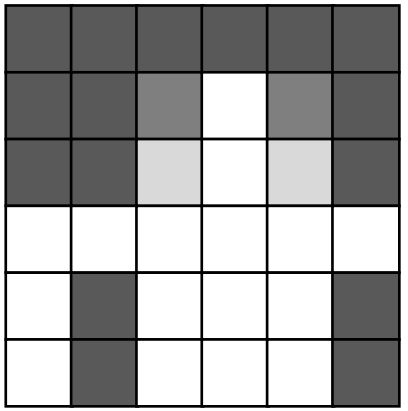
Afterwards, can add a small neural network that classifies the box and is applied to each window.

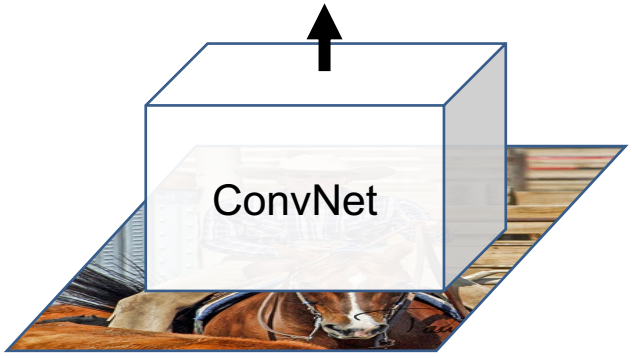
Mask RCNN

Feature Map
(e.g., 6x8x256)

Resize to fixed size (e.g., 7x7)
Details **critical**, but beyond scope of class.





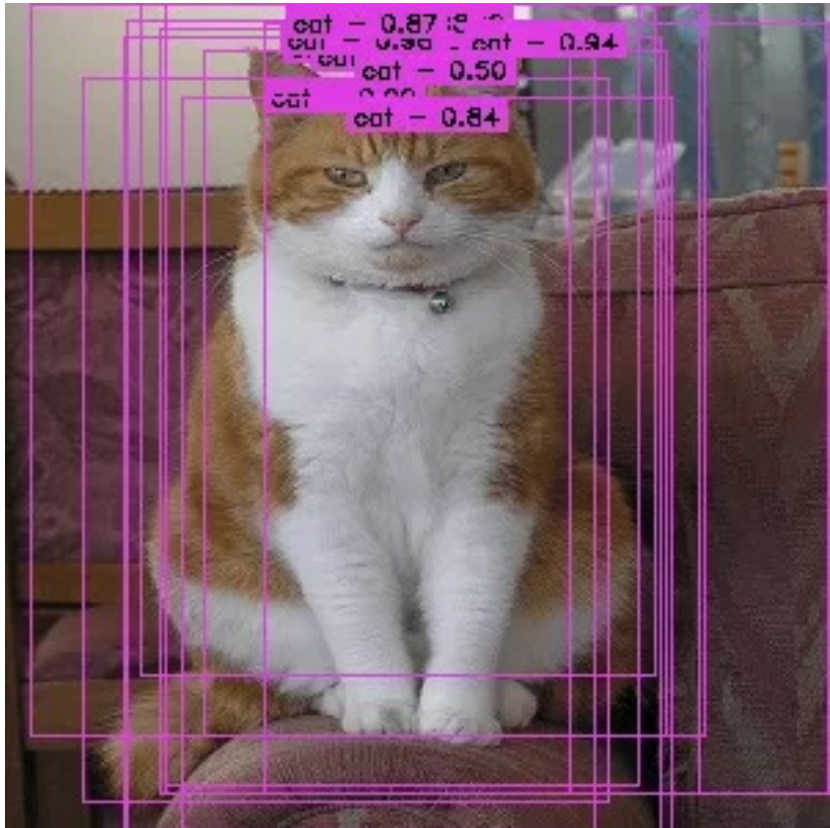


Can also predict a mask!
Everything is learned together
Simple and effective; details critical

Mask RCNN: He, Gkioxari, Dollar, Girshick . ICCV 2017.

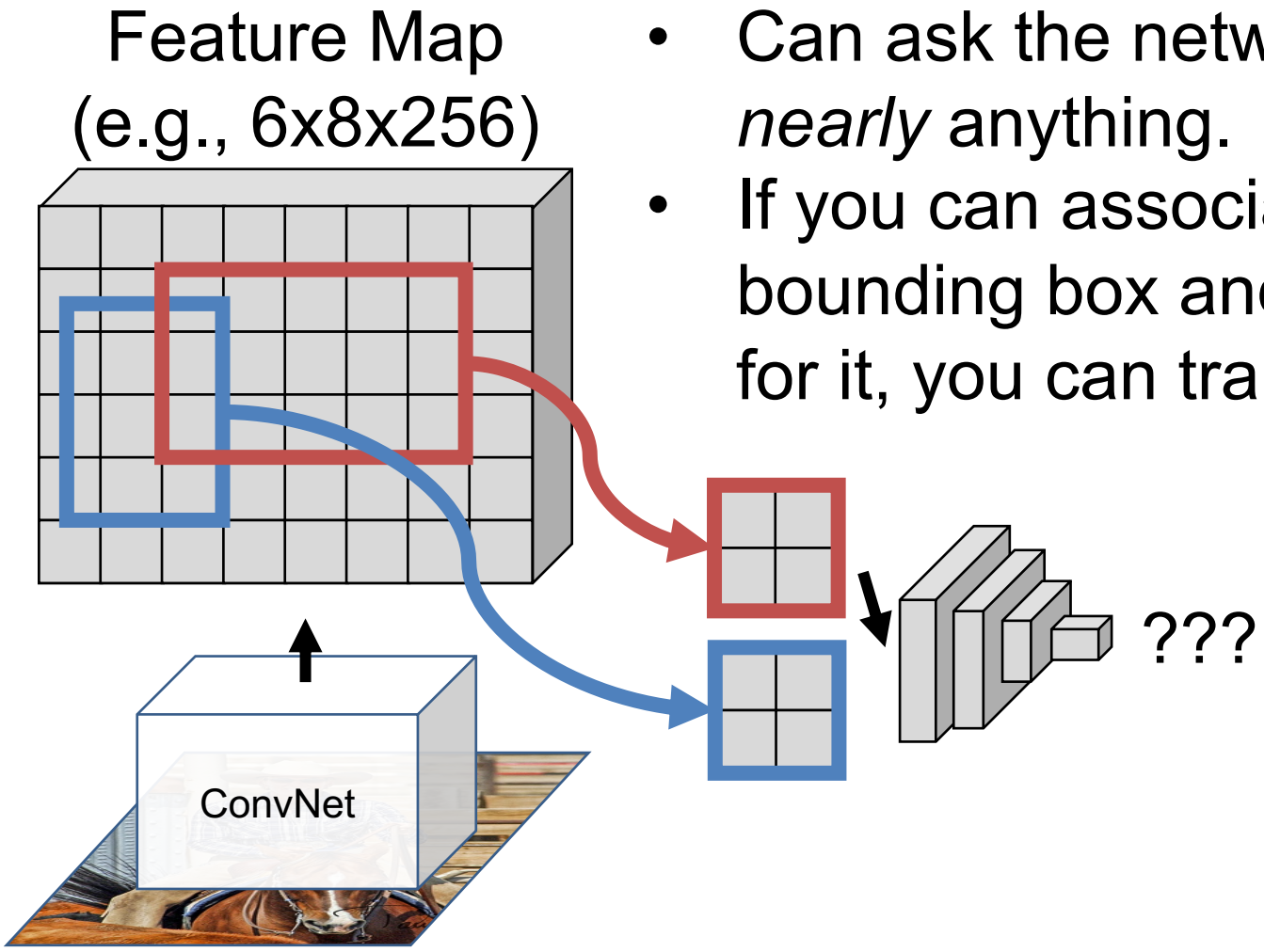
Non-Maximum Supression

1. Select ROI with highest confidence
2. Discard all ROIs with significant overlap (IoU)
3. Go to 1.



MaskRCNN – Results

Extending Object Detection



- Can ask the network to predict *nearly* anything.
- If you can associate it with a bounding box and can get data for it, you can train the model

Liu et al. *PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image.* CVPR 2019.

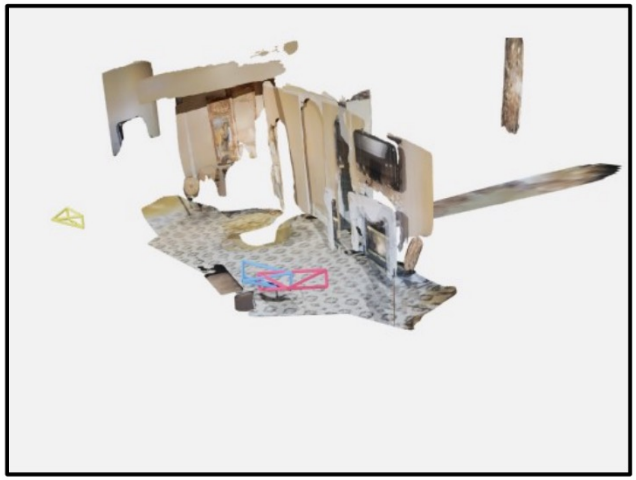
Extending Object Detection

Example: RGB image input, detect planar surfaces

Liu et al. *PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image.* CVPR 2019.

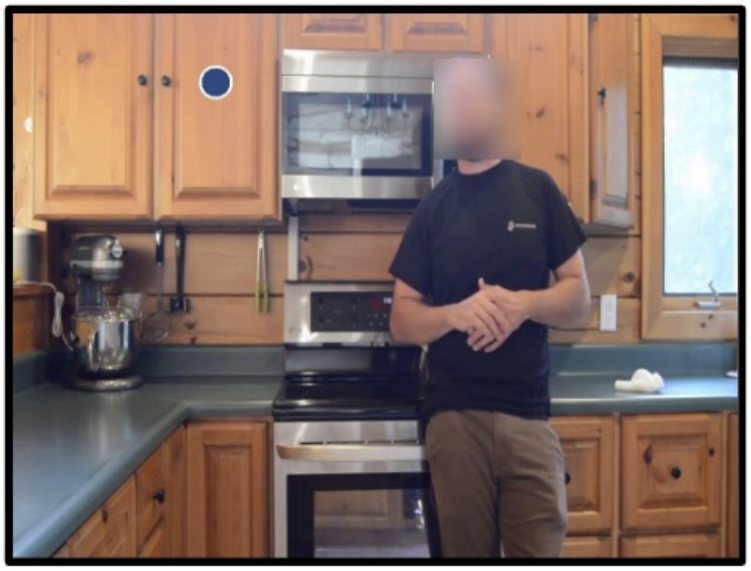
Extending Object Detection

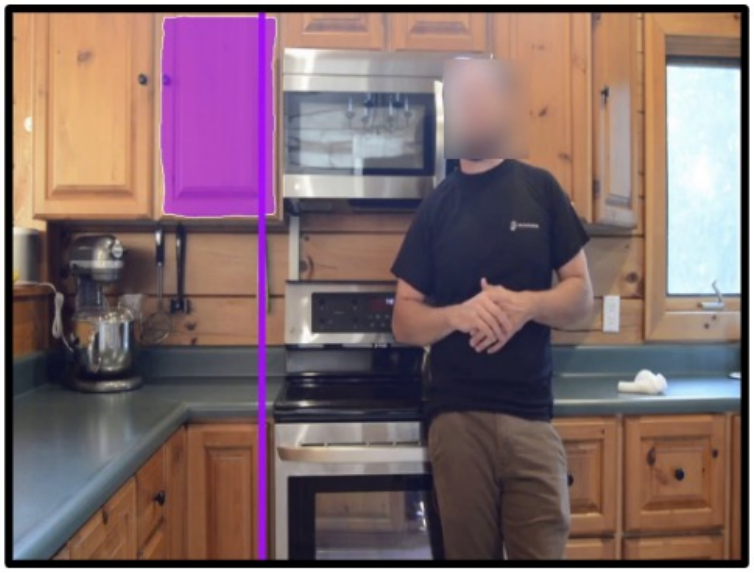
Core building block is detecting plane in image.

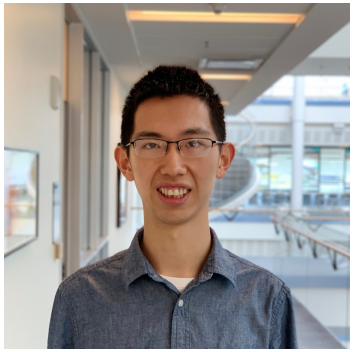


S. Agarwala, L. Jin, C. Rockwell, D.F. Fouhey. PlaneFormers: From Sparse View Planes to 3D Reconstruction.???, 2022

Other Fun Stuff

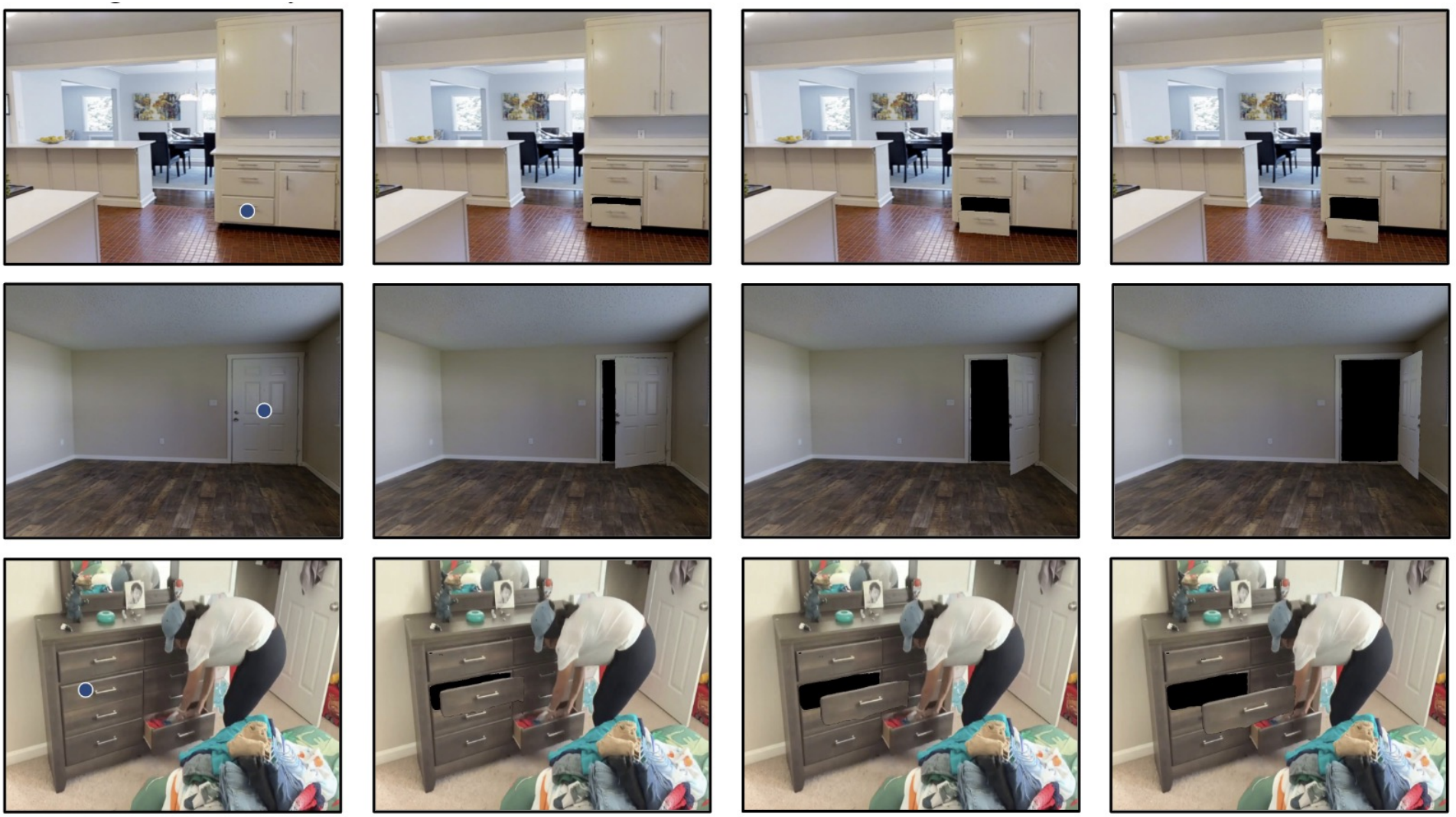




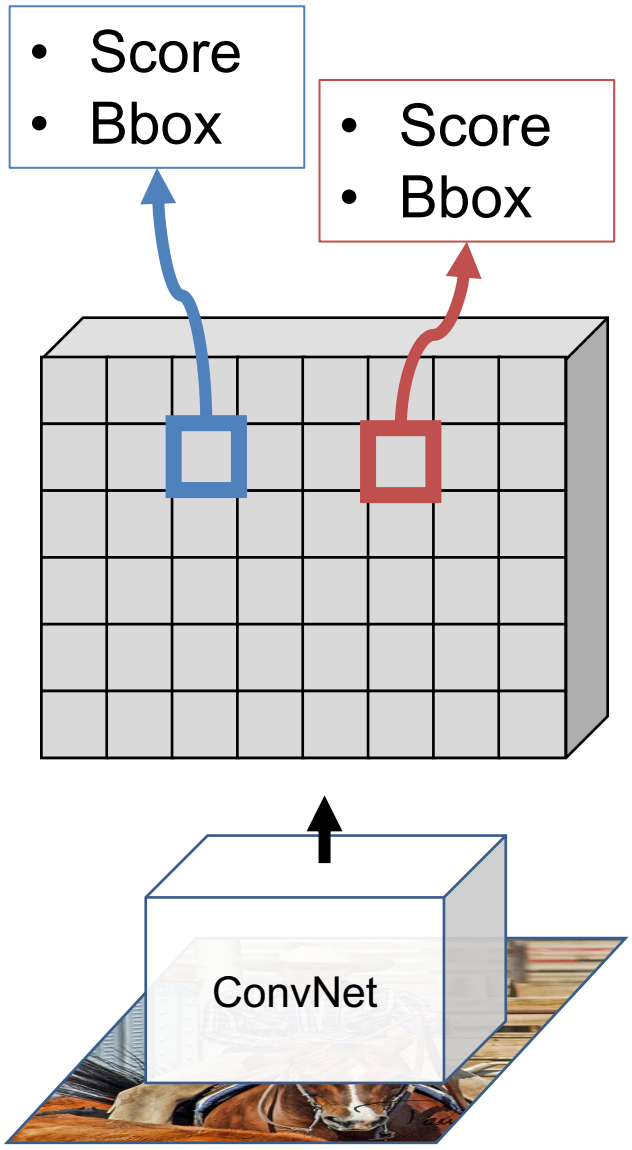


Shengyi Qian
Former 442 IA!

Other Fun Stuff



YOLO

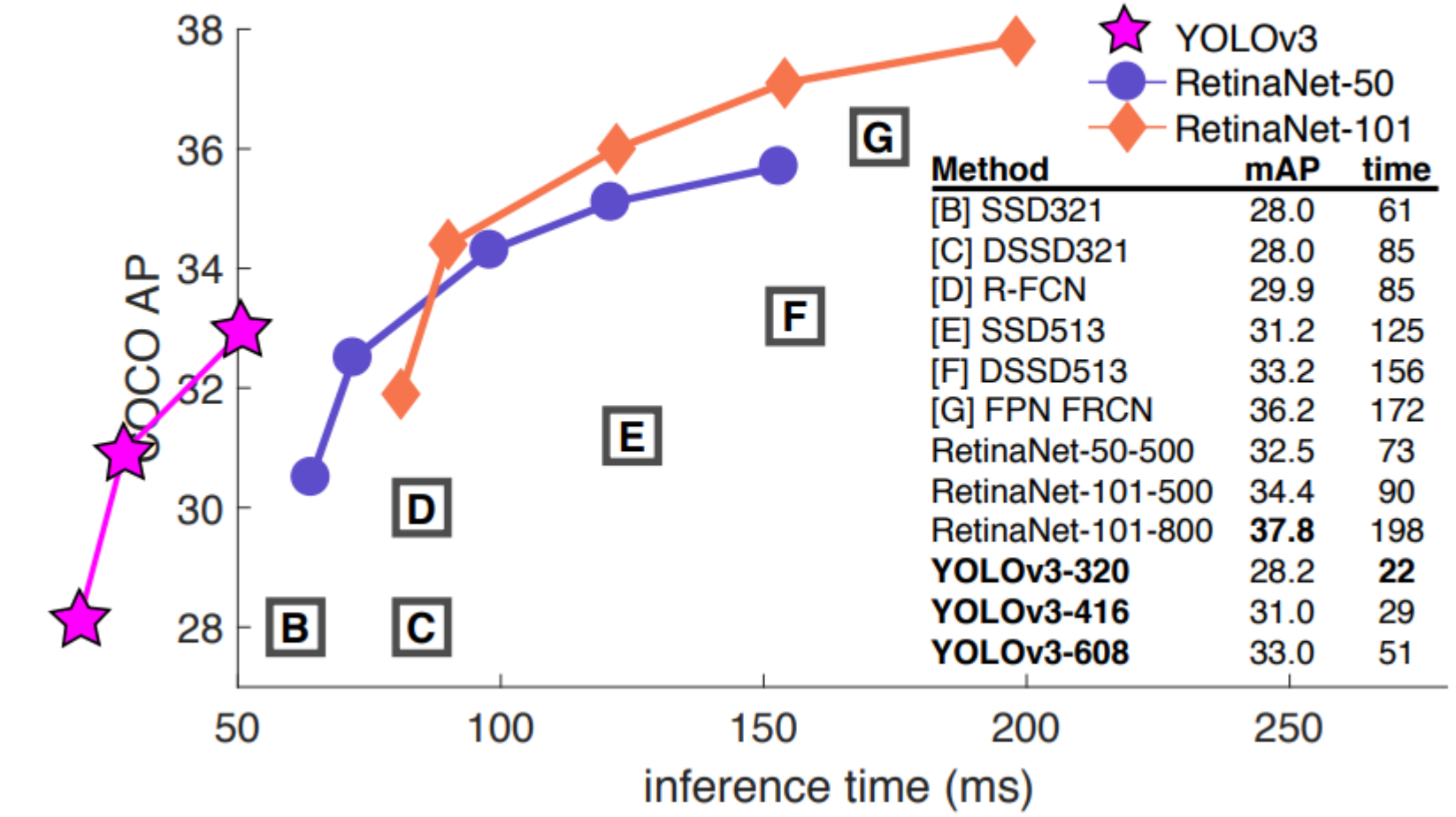


- No proposals
- Predict at each location in 7x7 feature map, score for each class + 2 bboxes
- 7x faster than Faster-RCNN, but worse accuracy, precision
- Immensely popular in robotics
- Loads of similar methods (YOLOv2, YOLOv3)

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016

YOLO

- Score



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

J. Redmon, A. Farhadi, *YOLOv3: An Incremental Improvement .*

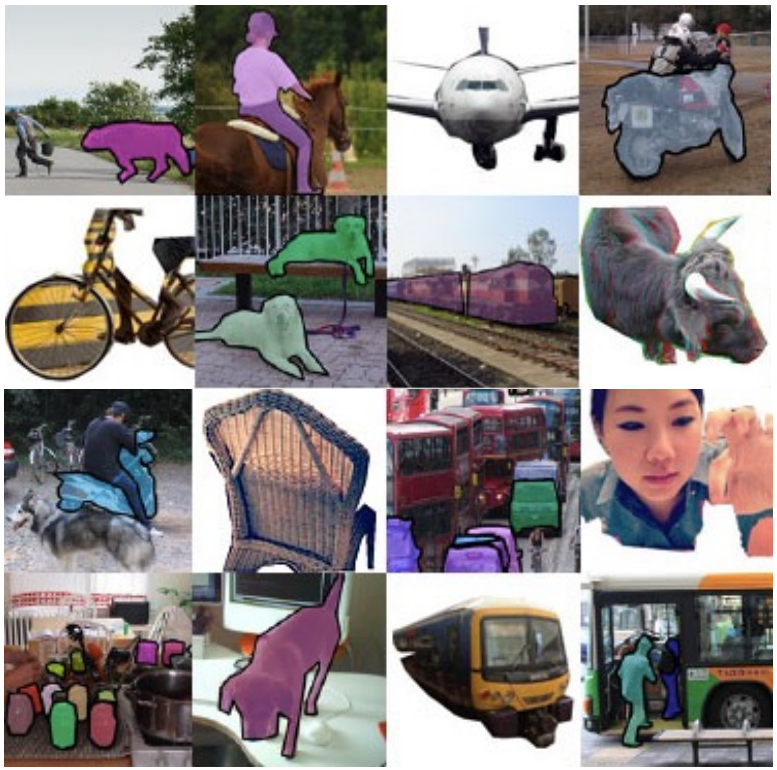
DEMO

New detection benchmark: COCO (2014)

- 80 categories instead of PASCAL's 20
- Current best mAP: 66%

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints



http://cocodataset.org/#home

A Few Caveats

- Flickr images come from a really weird process
- Step 1: user takes a picture
- Step 2: user decides to upload it
- Step 3: user decides to write something like “refrigerator” somewhere in the description
- Step 4: a vision person stumbles on it while searching Flickr for refrigerators for a dataset

Who takes photos of open refrigerators ?????

Kitchens from Googling










Places 365 Dataset, Zhou et al. ‘17

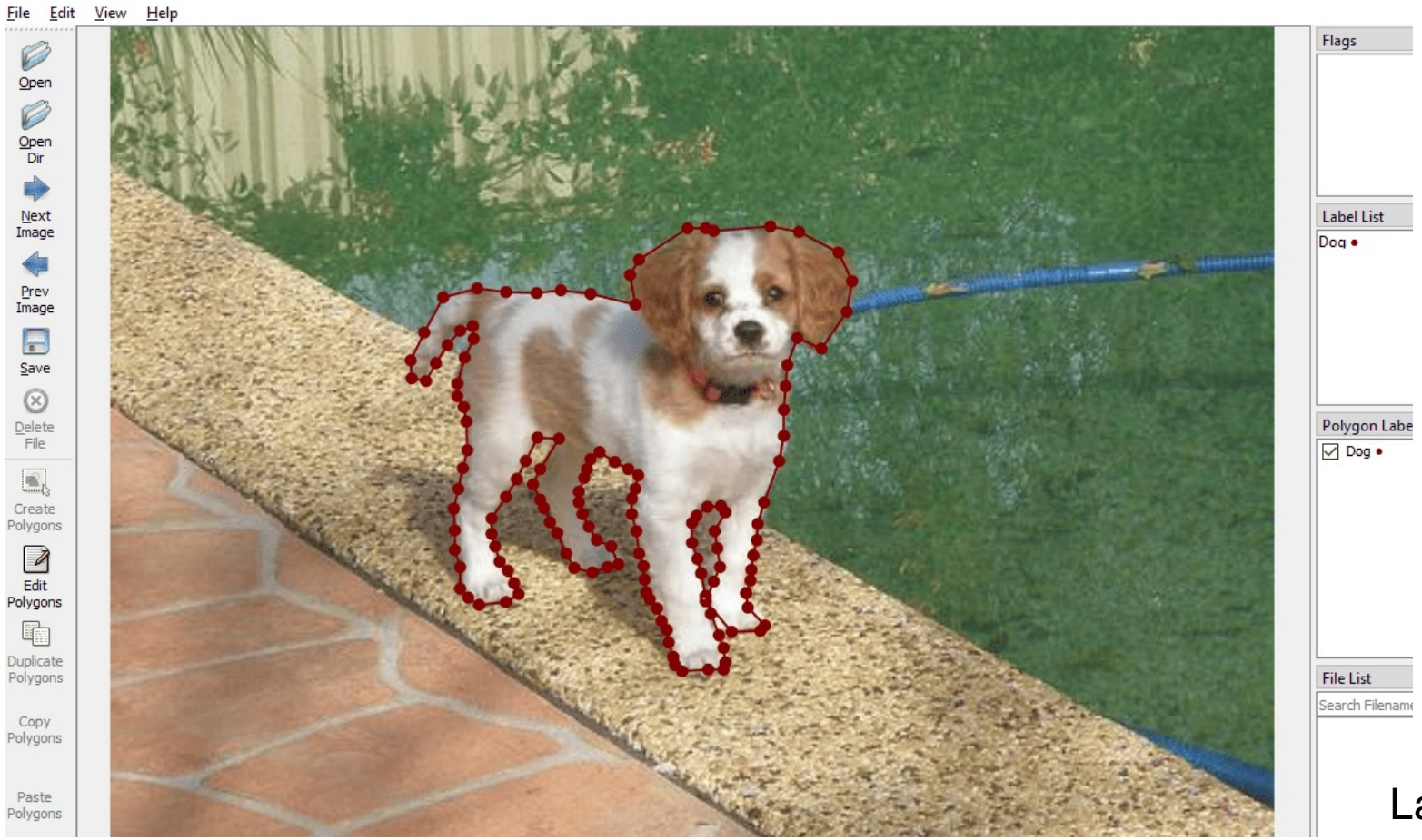
Guess the category!

These were detected with >90% confidence

(1) Person (2) Giraffe (3) Bicycle

Self-Supervised Feature Learning

- Can we train without expensive labeling?



LabelMe

Y. LeCun

How Much Information is the Machine Given during Learning?

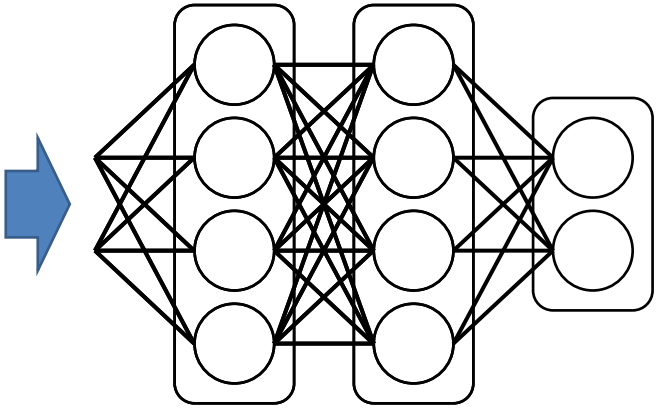
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann Lecun

Transfer Learning

- w = **weights_from_somewhere_else**
- for batch in batches:
  - inputs, labels = batch
  - calculate gradient of loss function with respect to w applied to samples in inputs
  - w += gradient

ImageNet + Deep Learning

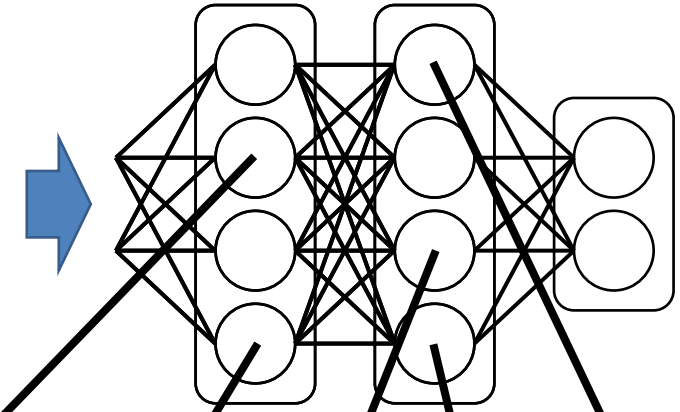


Beagle

- Image Retrieval
- Detection
- Segmentation
- Depth Estimation
- …

Slide Credit: C. Doersch

ImageNet + Deep Learning



Beagle

Materials?

Parts?

Pose?

Geometry?

Boundaries?

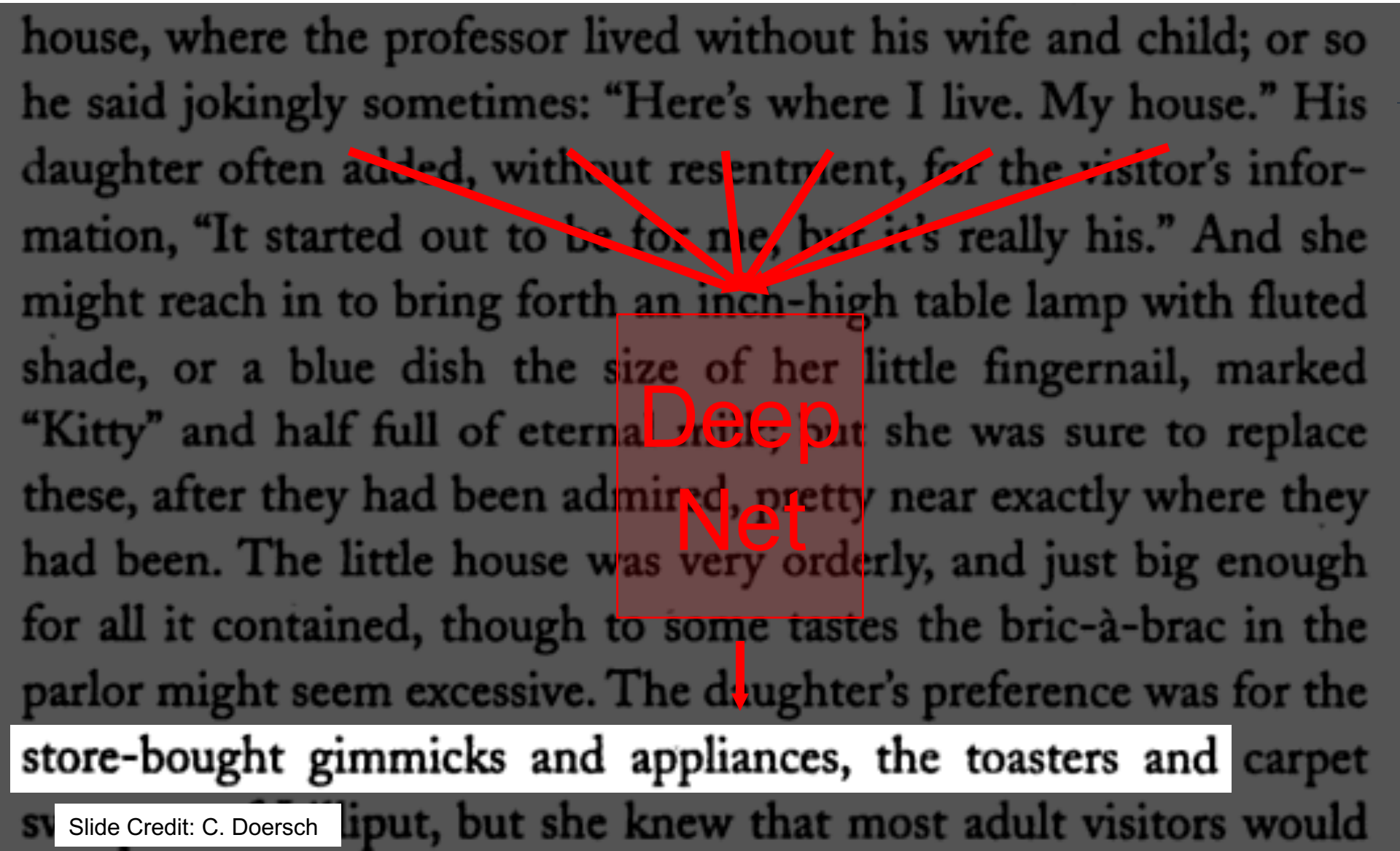
Do we even need semantic labels?

Do we need this task?

Slide Credit: C. Doersch

Context as Supervision

[Collobert & Weston 2008; Mikolov et al. 2013]

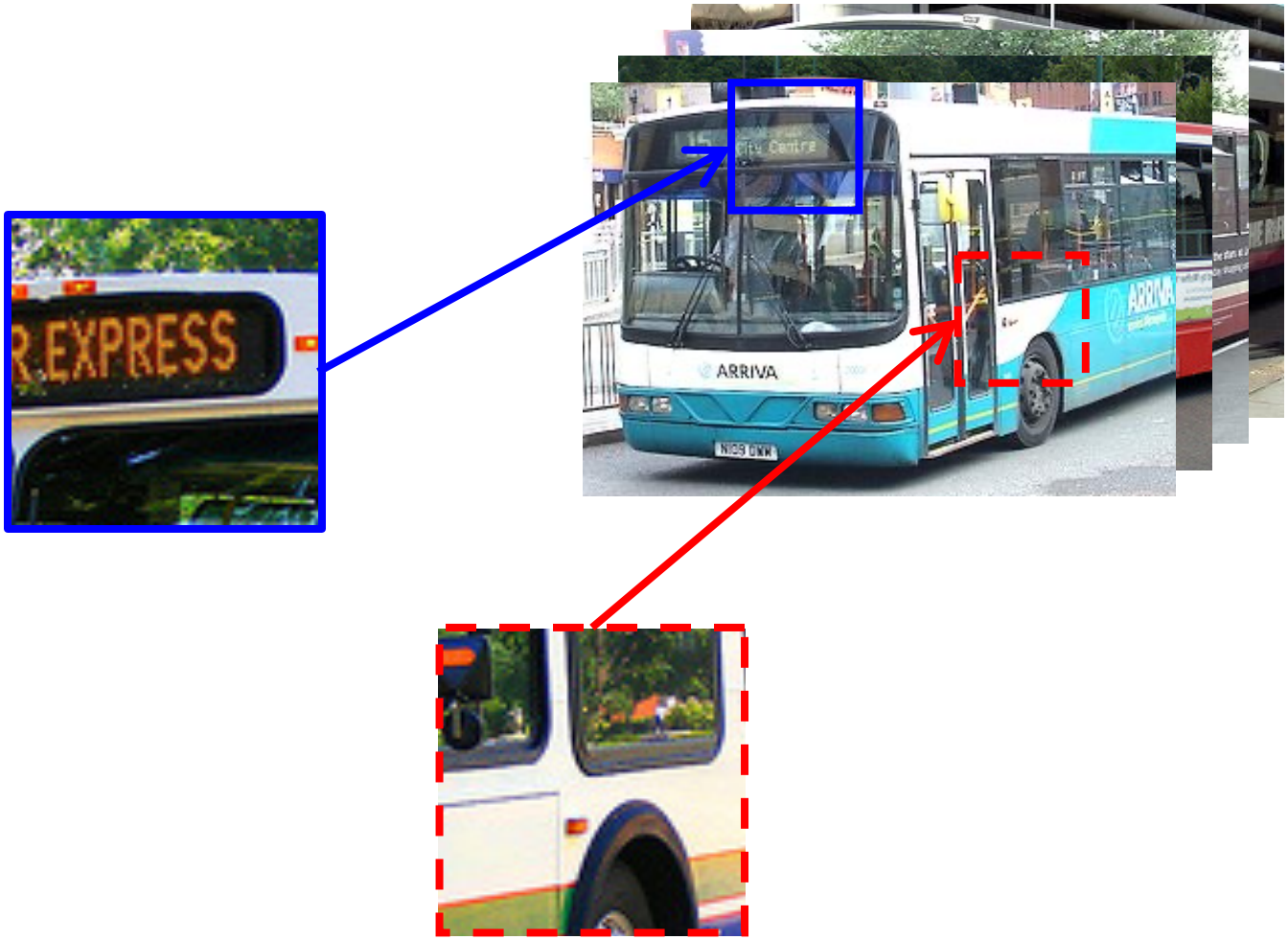


Slide Credit: C. Doersch

Context Prediction for Images

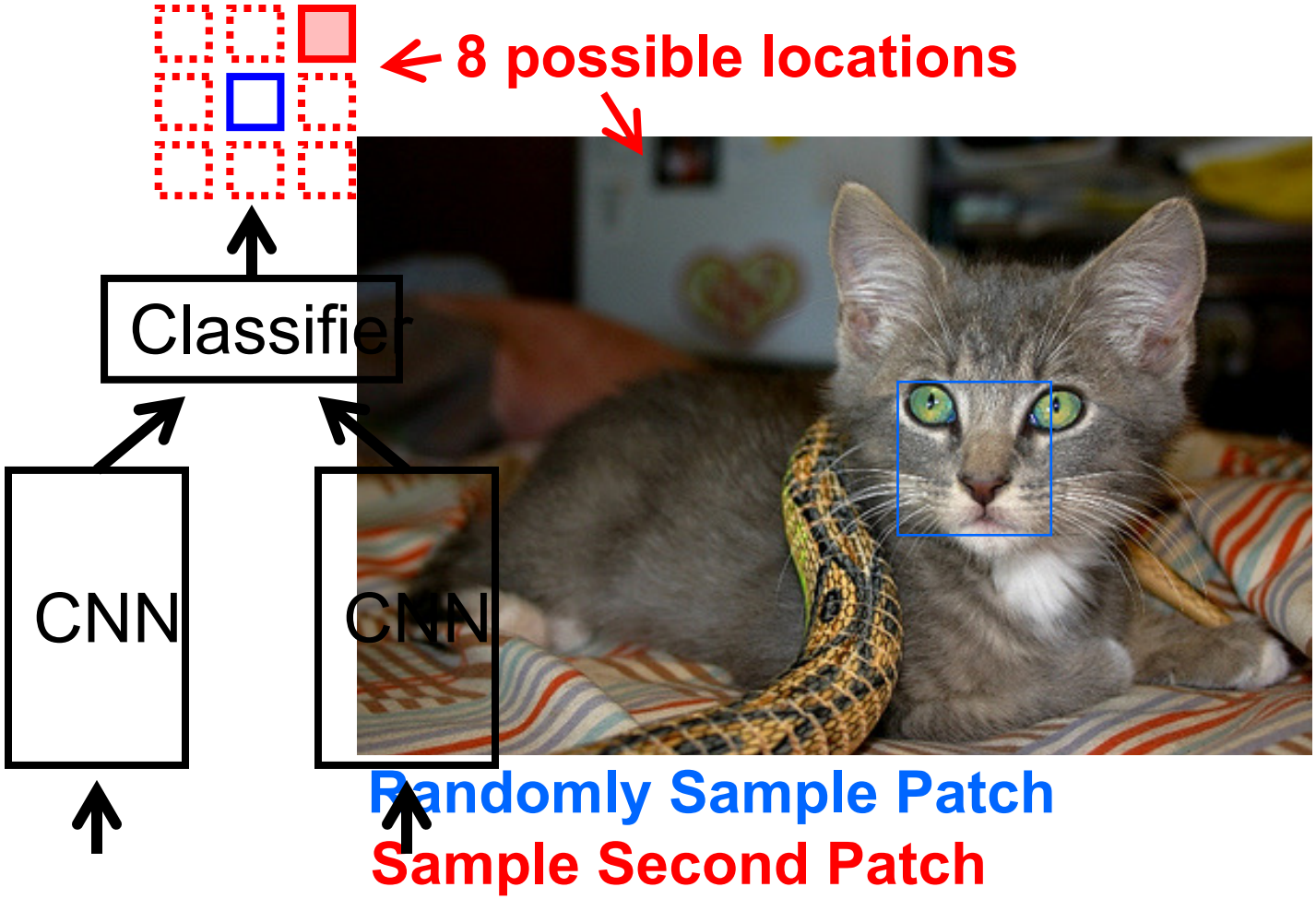
Slide Credit: C. Doersch

Semantics from a non-semantic task

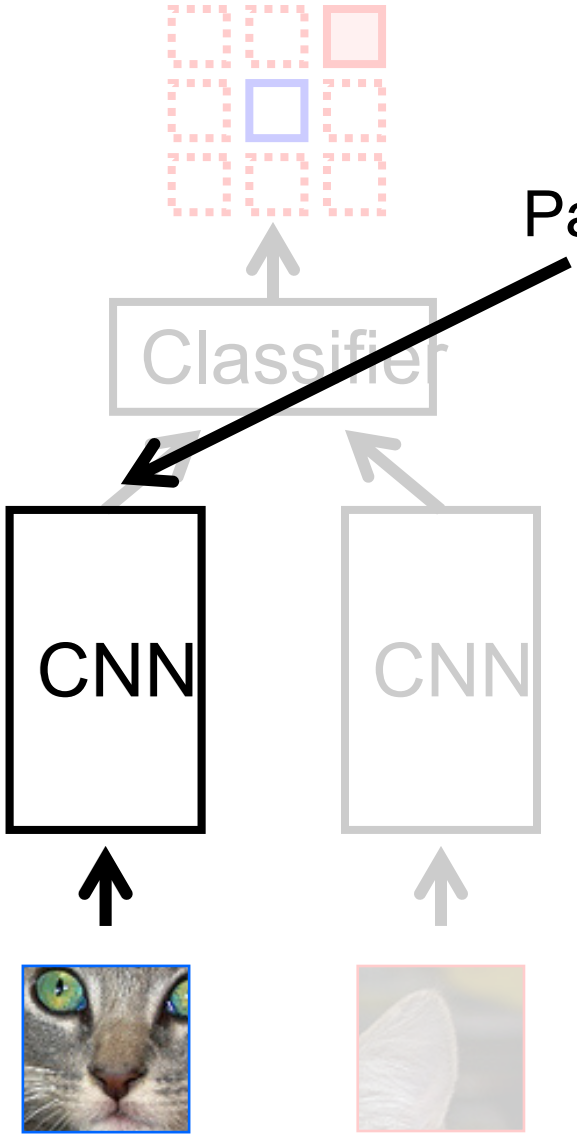


Slide Credit: C. Doersch

Relative Position Task



Slide Credit: C. Doersch



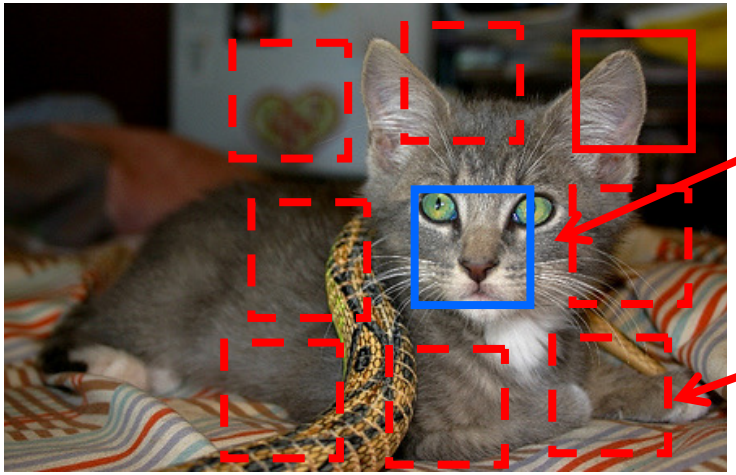
Input

Nearest Neighbors

Note: connects ***across*** instances

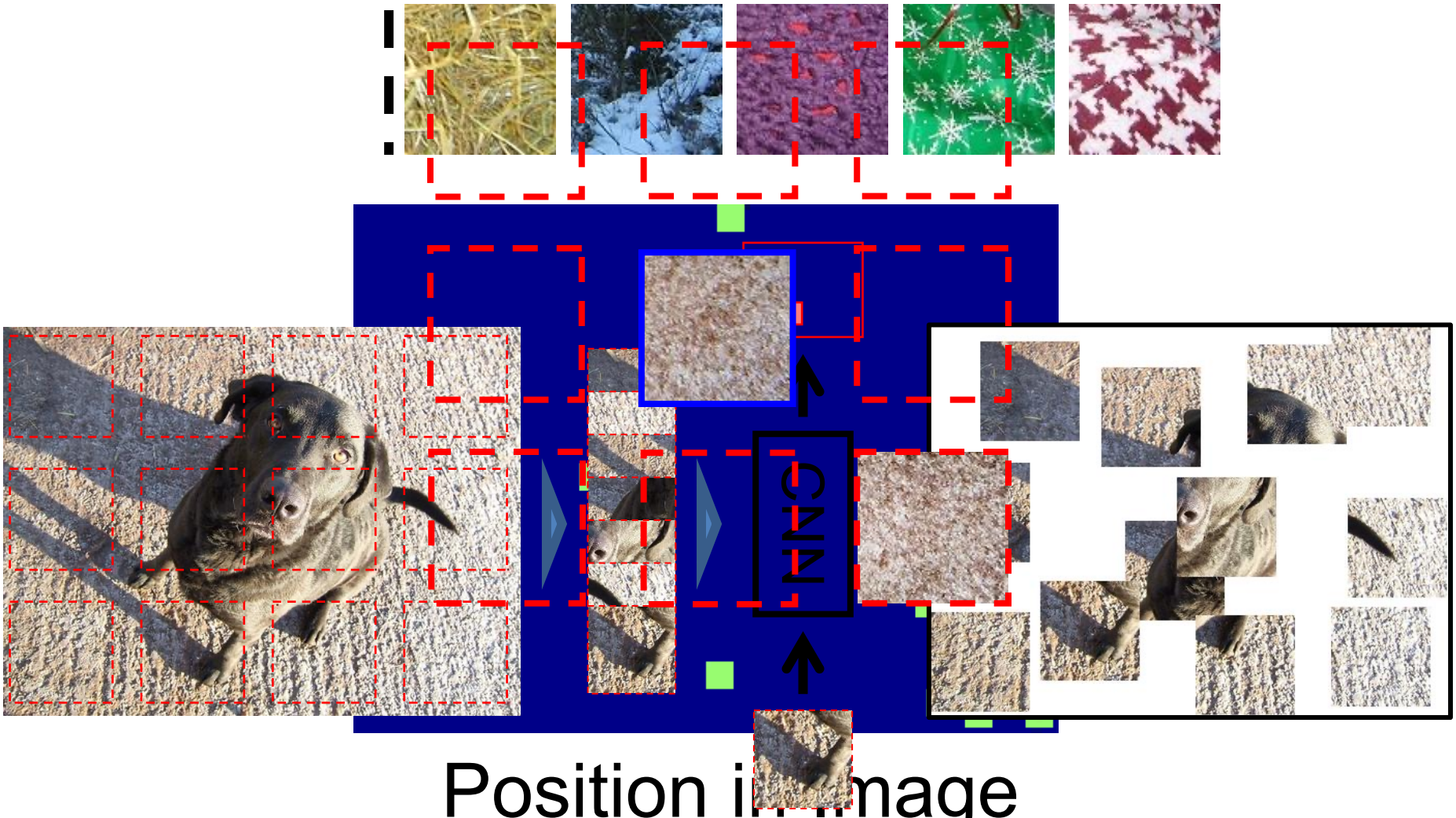
Slide Credit: C. Doersch

Avoiding Trivial Shortcuts



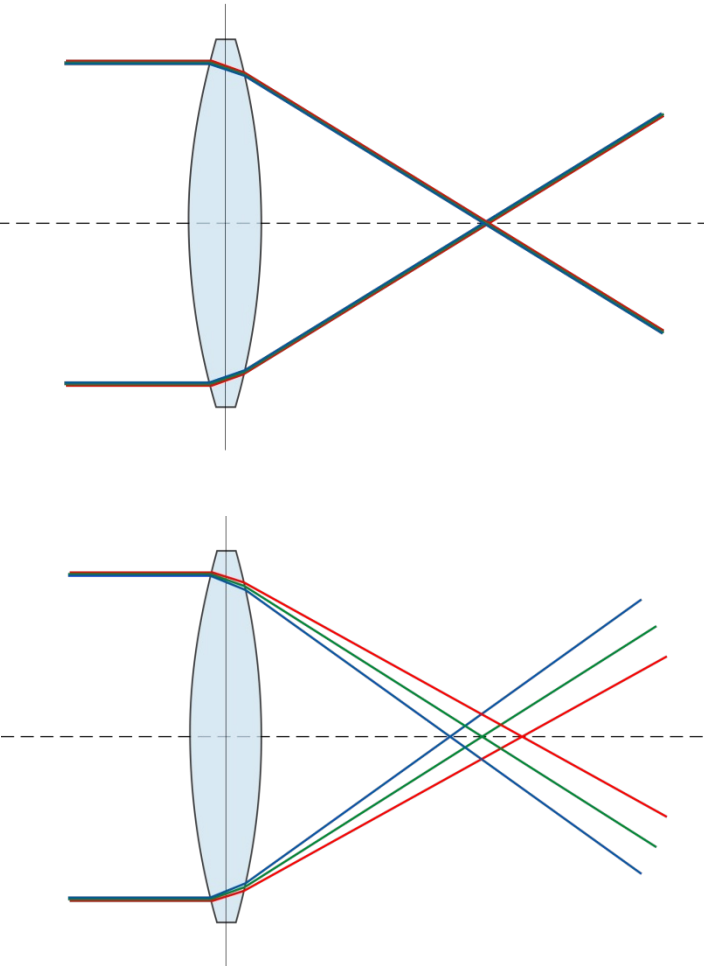
Slide Credit: C. Doersch

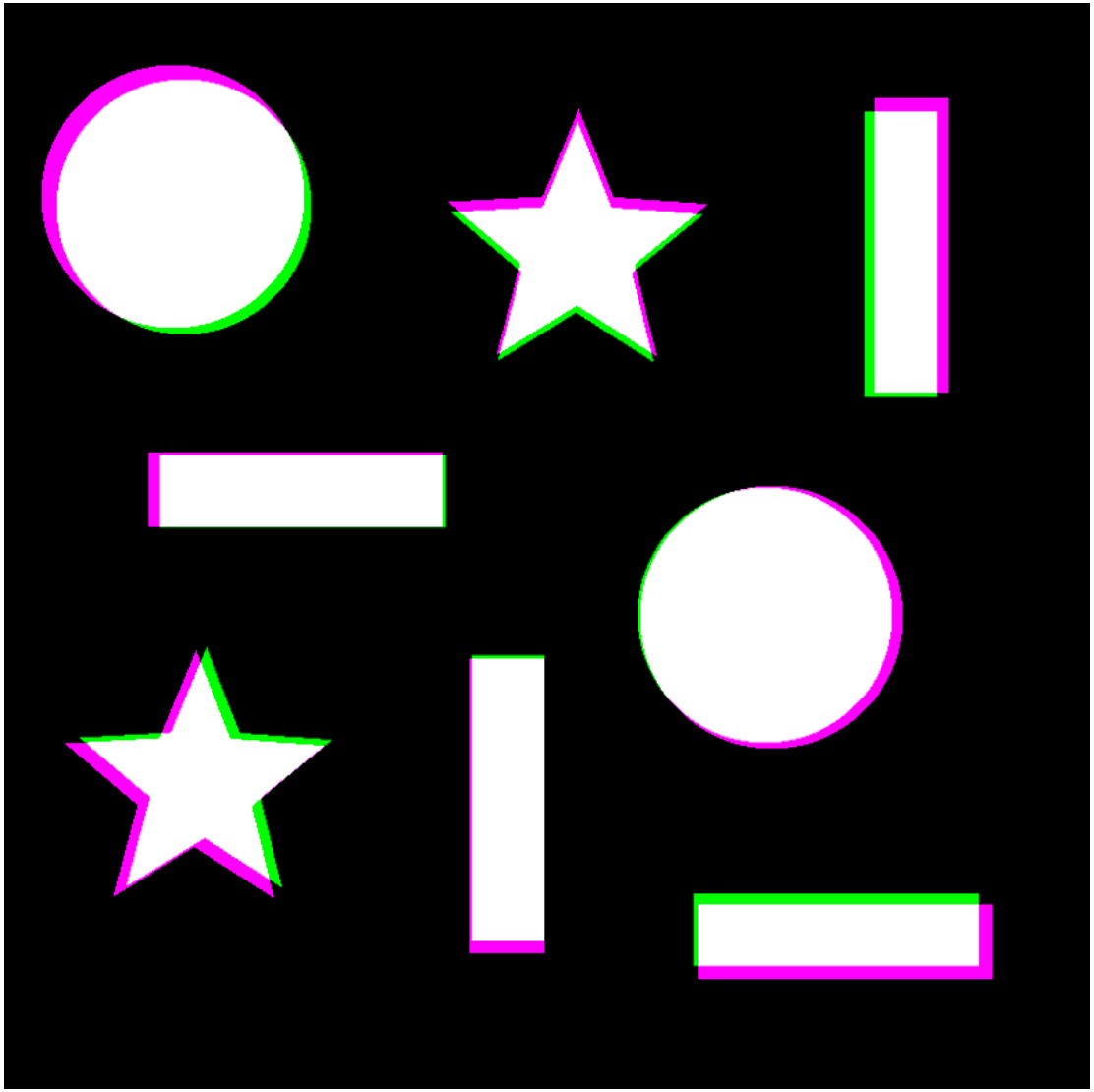
A Not-So “Trivial” Shortcut



Slide Credit: C. Doersch

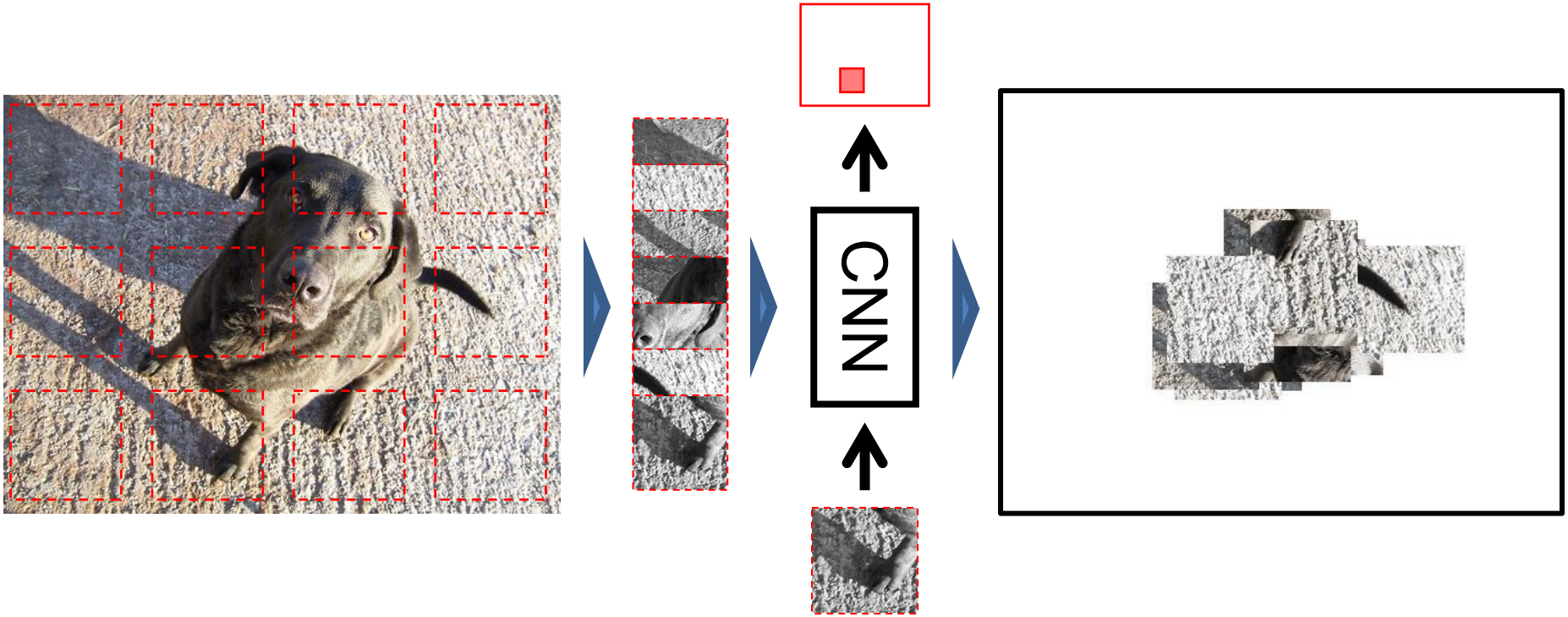
Chromatic Aberration





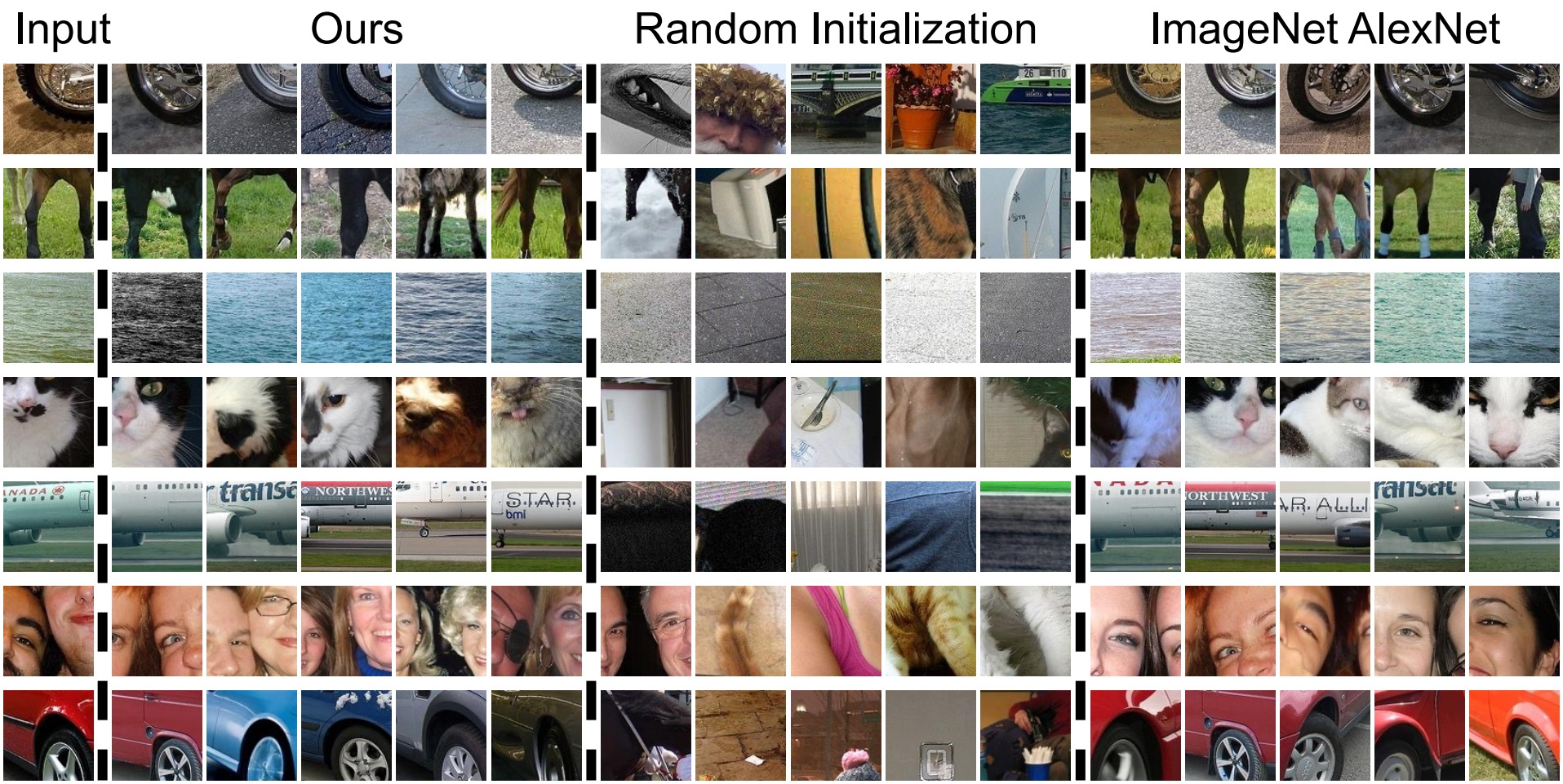
Slide Credit: C. Doersch

Chromatic Aberration



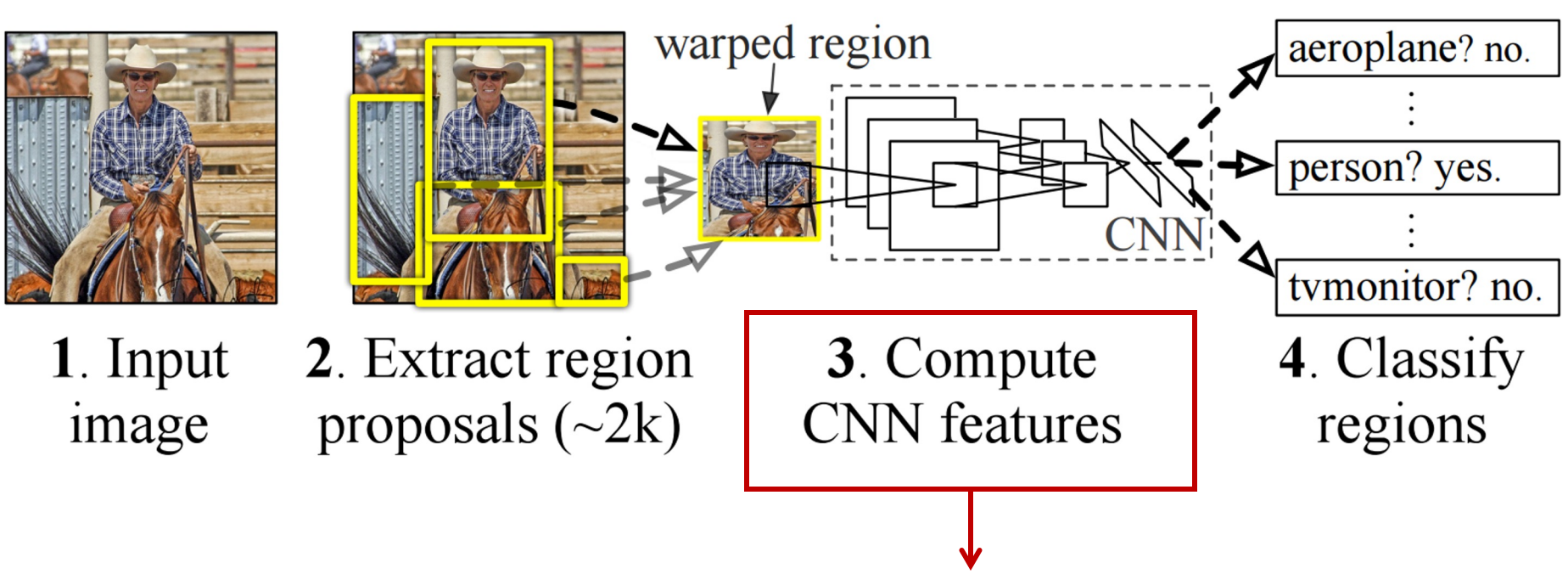
Slide Credit: C. Doersch

What is learned?



Slide Credit: C. Doersch

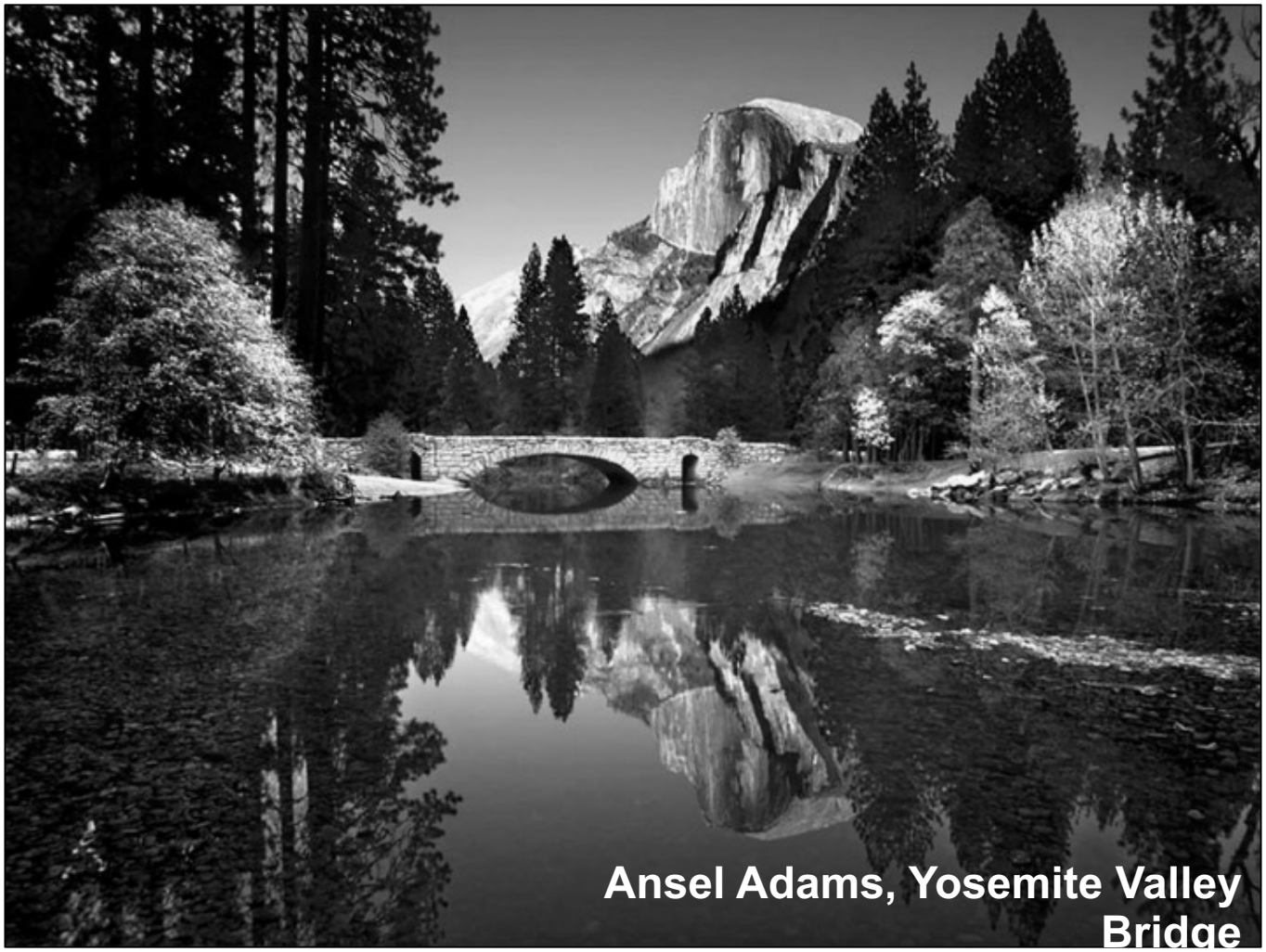
Pre-Training for R-CNN



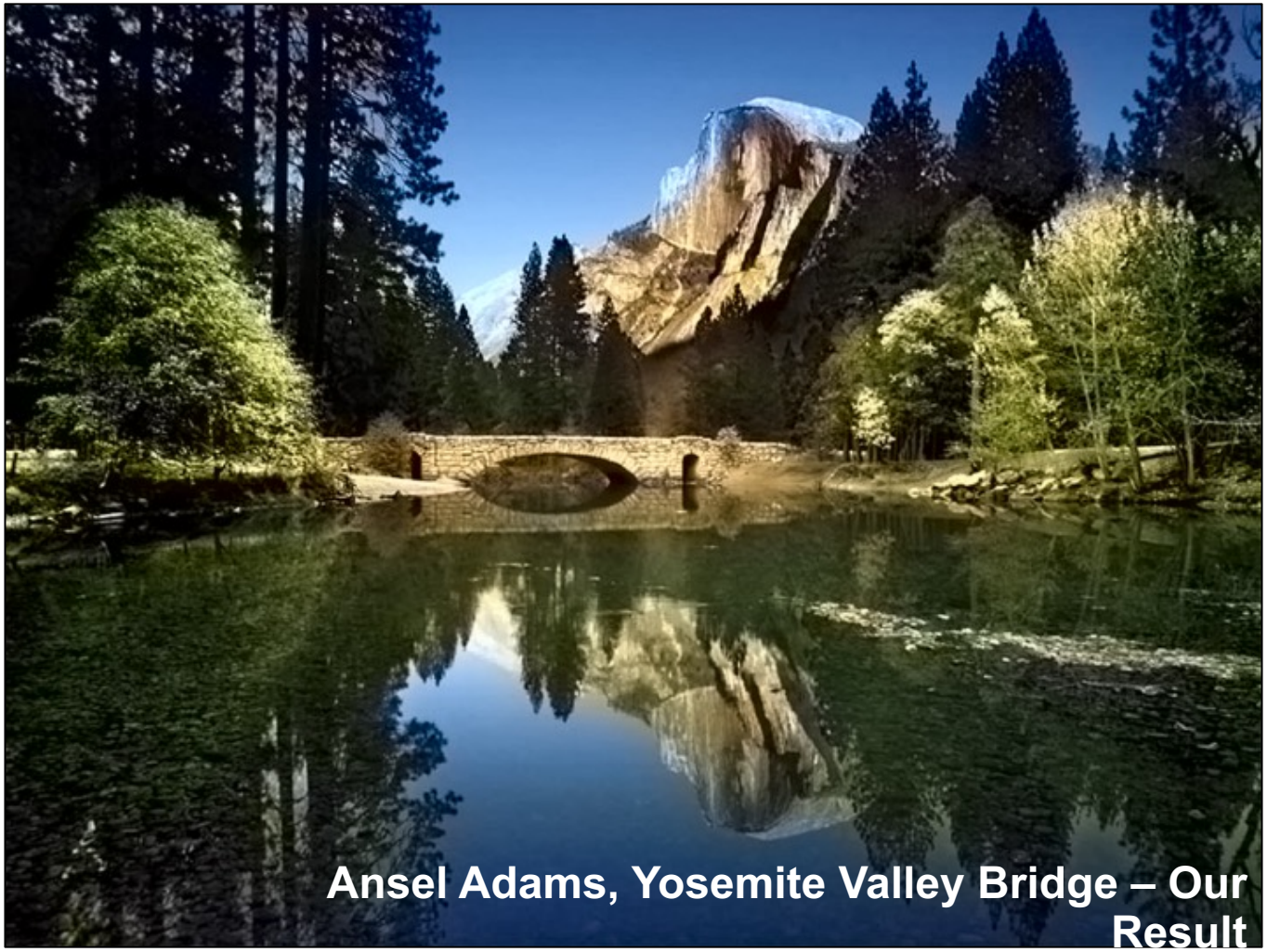
Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

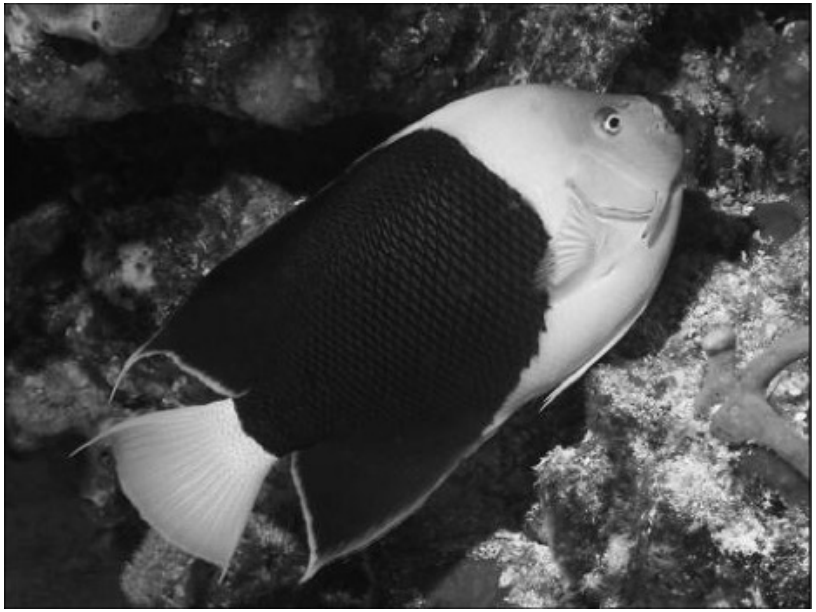
Other Sources Of Signal



Slide Credit: R. Zhang



Slide Credit: R. Zhang



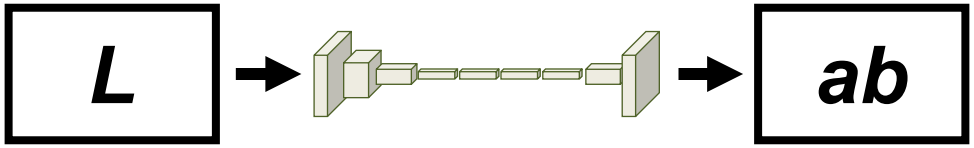
$\mathcal{F}$

Grayscale image: *L* channel

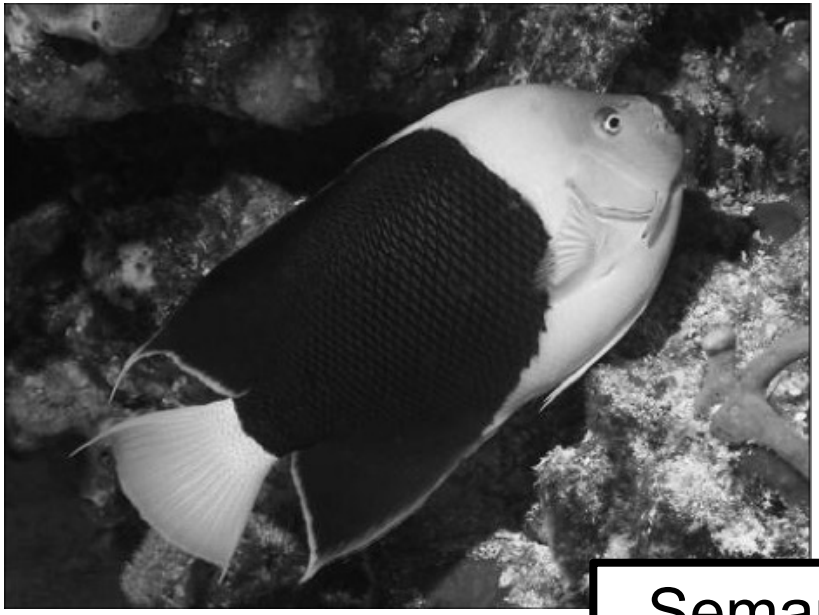
$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: *ab* channels

$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$



Slide Credit: R. Zhang



$\mathcal{F}$

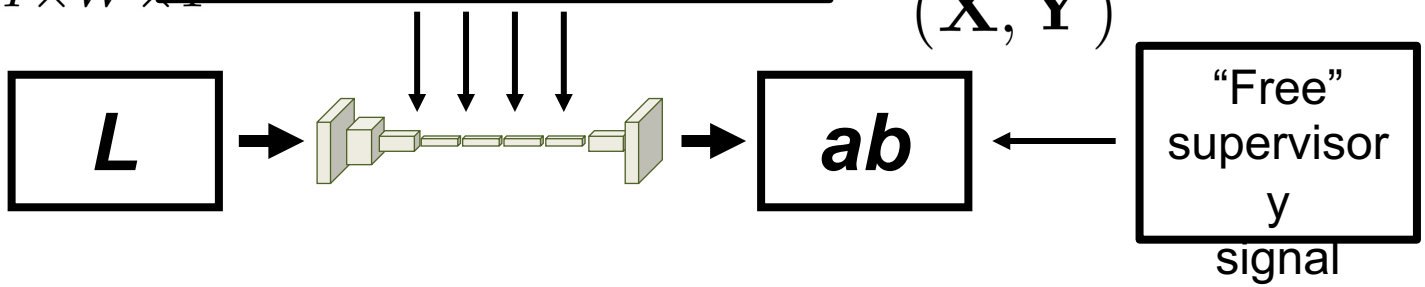
Semantics? Higher-level abstraction?

Grayscale image: *L*

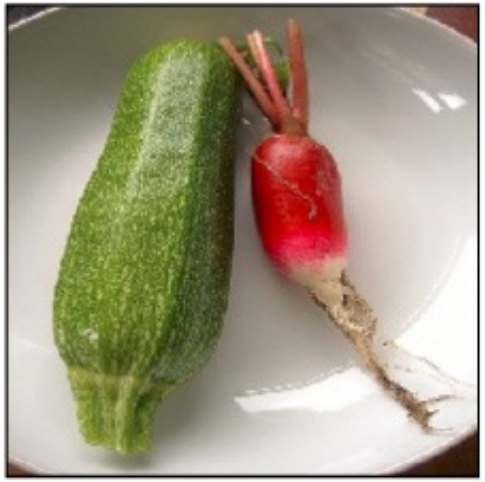
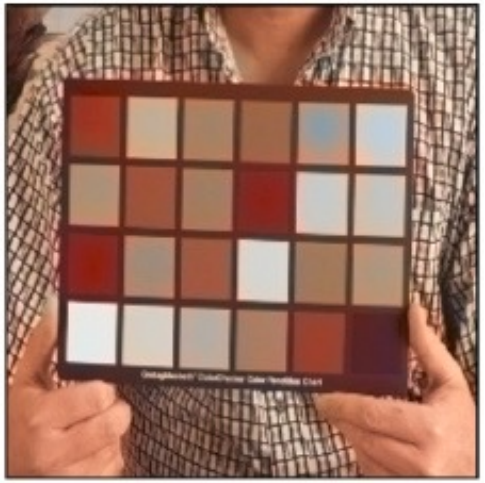
$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

oncatenate (L,ab)

$(\mathbf{X}, \widehat{\mathbf{Y}})$

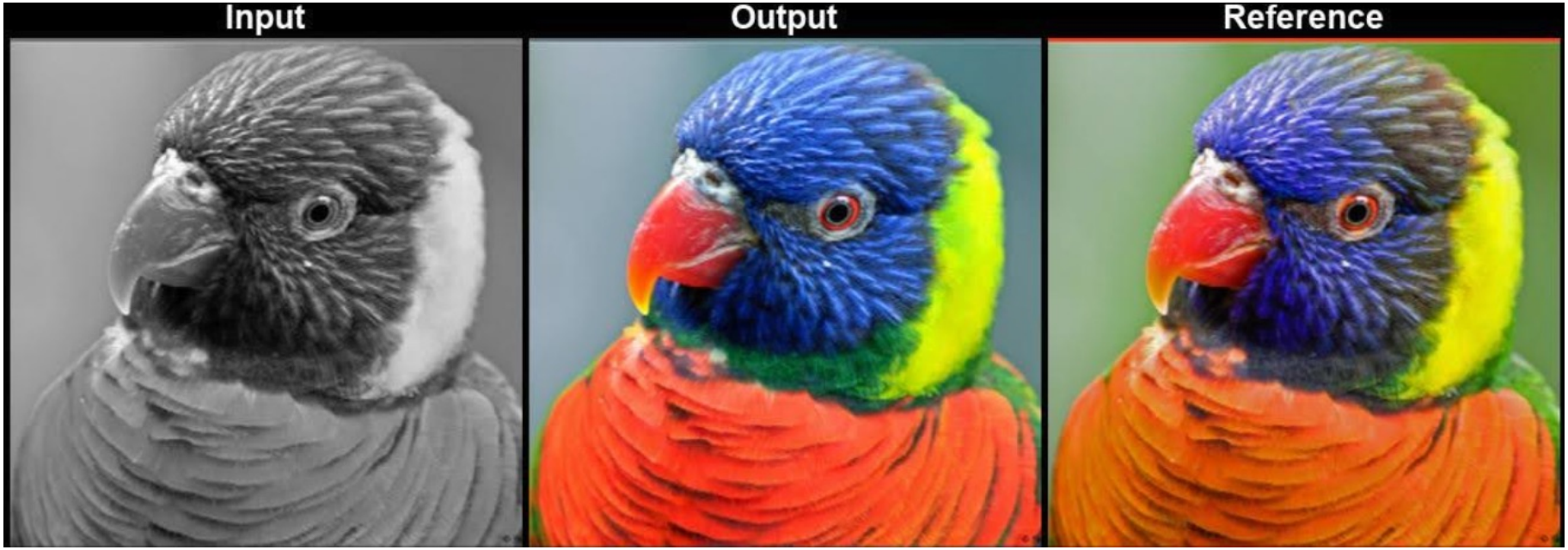


Slide Credit: R. Zhang

Input	Ground Truth	Output
		
		

Slide Credit: R. Zhang

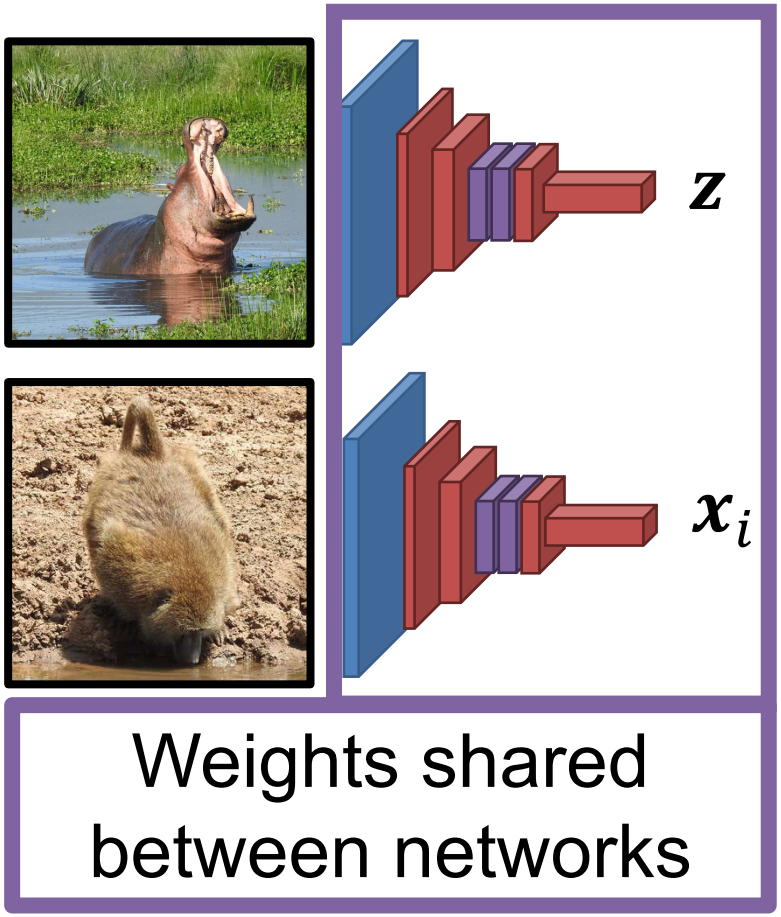
Got Much Better with Diffusion Models



C. Saharia et al. Palette


Visually Indicated Sounds
Andrew Owens
Phillip Isola
Josh McDermott
Antonio Torralba
Edward Adelson
William Freeman

Contrastive Learning



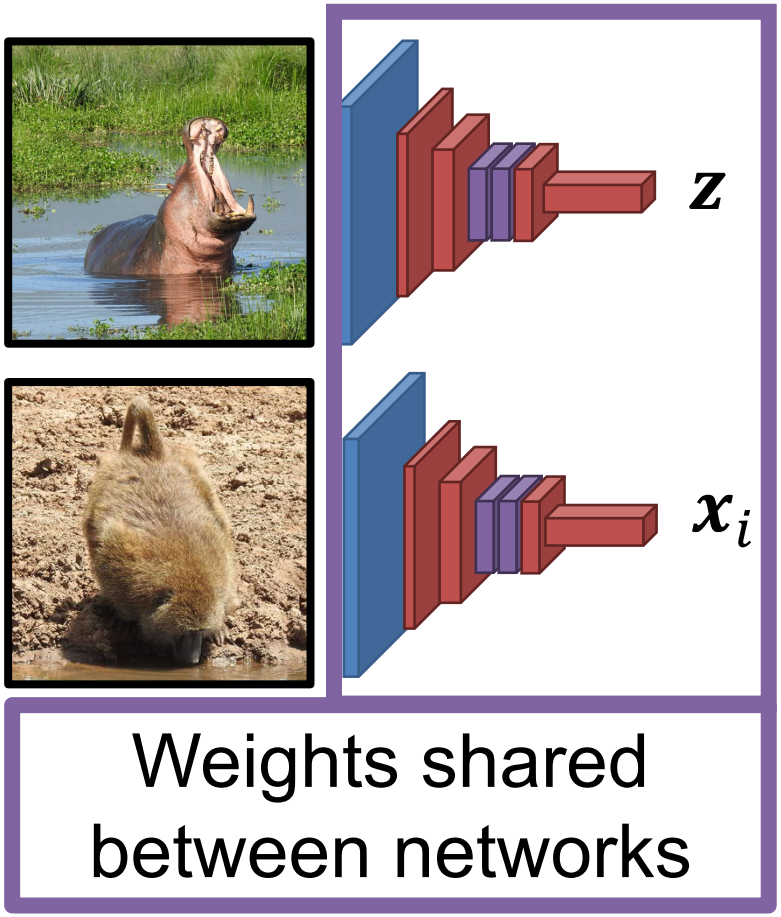
Given sample, construct feature z; take a bunch of other images, minimize:

$$-\log\left(\frac{\exp(\boldsymbol{z}^T\boldsymbol{z})}{\exp(\boldsymbol{z}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{z}^T\boldsymbol{x})}\right)$$

$$\frac{\exp(\boldsymbol{z}^T\boldsymbol{z})}{\exp(\boldsymbol{z}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{z}^T\boldsymbol{x})} \leq 1$$

He et al. Momentum Contrastive Learning, 2019. Wu et al. Instance discrimination 2018

Contrastive Learning



Given sample, construct feature z; take a bunch of other images, minimize:

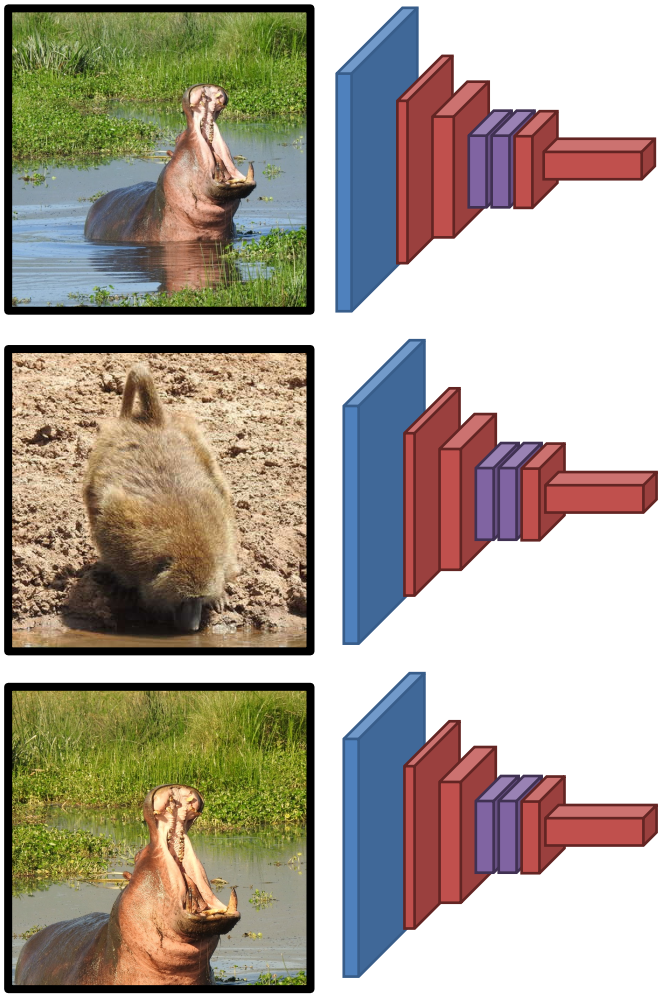
$$-\log\left(\frac{\exp(\boldsymbol{z}^T\boldsymbol{z})}{\exp(\boldsymbol{z}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{z}^T\boldsymbol{x})}\right)$$

Basically, a scoring function with w = z to measure similarity:

$$\frac{\exp(\boldsymbol{w}^T\boldsymbol{z})}{\exp(\boldsymbol{w}^T\boldsymbol{z})+\sum_i \exp(\boldsymbol{w}^T\boldsymbol{x})}$$

He et al. Momentum Contrastive Learning, 2019. Wu et al. Instance discrimination 2018

Contrastive Learning



z

x_i

z'

Best performing methods measure distance to augmented sample z':

$$-\log\left(\frac{\exp(z^T z')}{\exp(z^T z') + \sum_i \exp(z^T x)}\right)$$

Goal: score augmented sample higher than everything else.

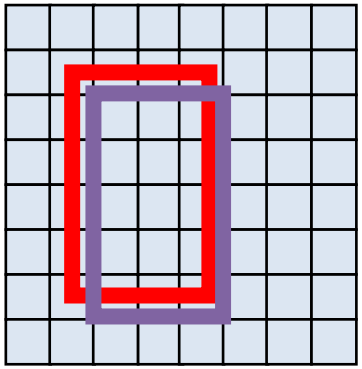
He et al. Momentum Contrastive Learning, 2019. Wu et al. Instance discrimination 2018

Next Time

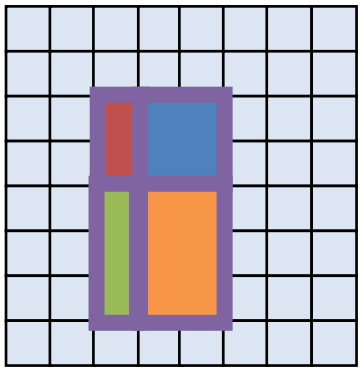
- Synthesizing Images

Extra Stuff

Fast R-CNN – ROI-Pool

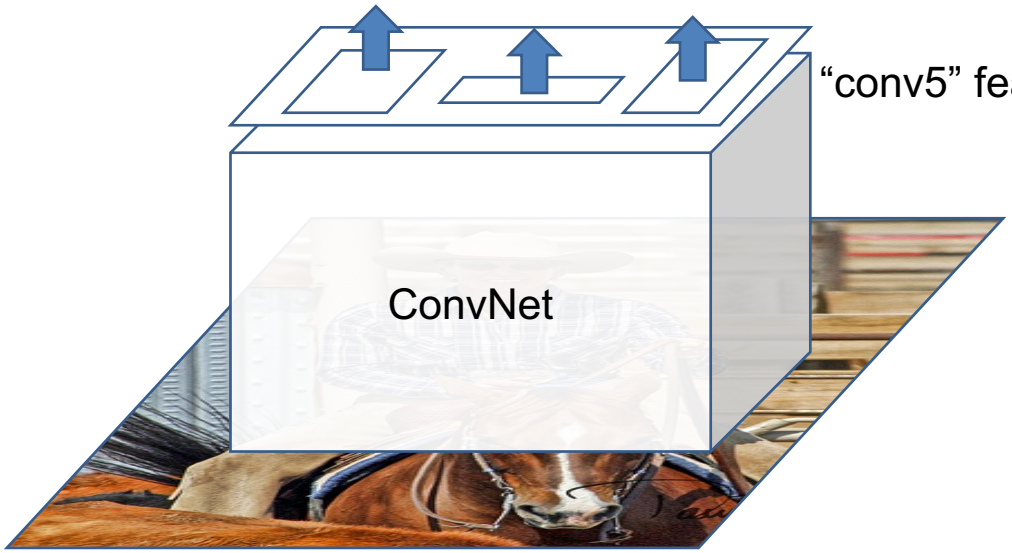


Line up



Divide

Pool

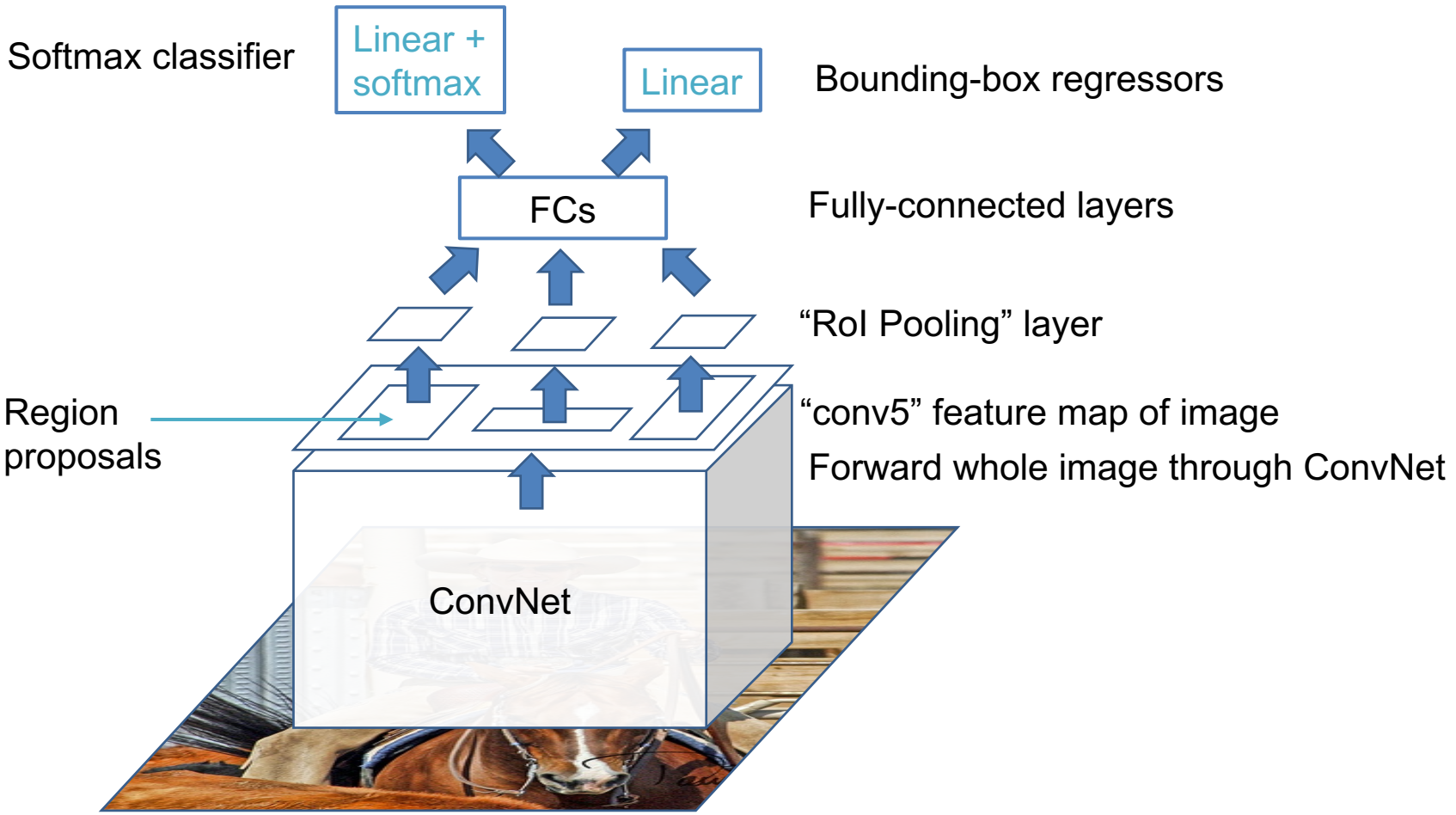


"conv5" feature map of image

Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015

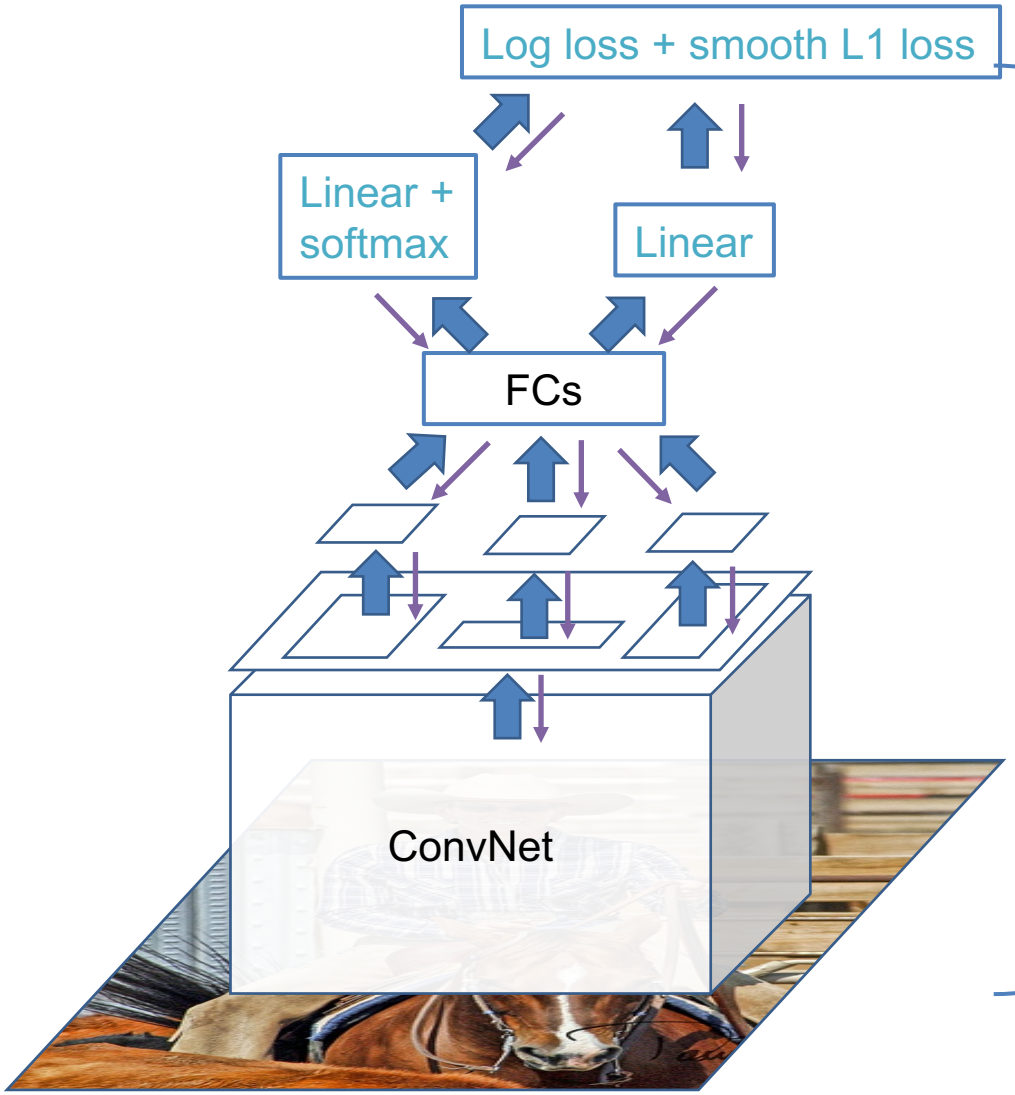
Fast R-CNN



Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015

Fast R-CNN training



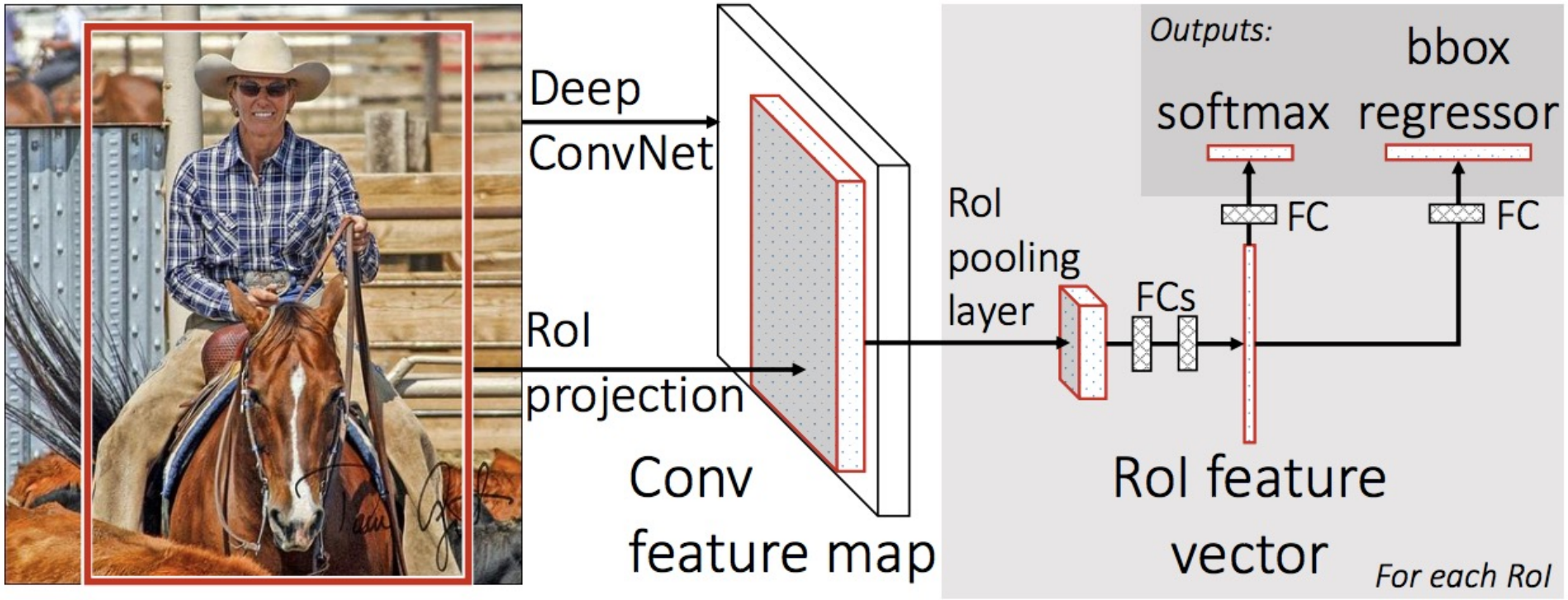
Multi-task loss

Trainable

Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015

Fast R-CNN: Another view



Source: R. Girschick

R. Girshick, Fast R-CNN, ICCV 2015

Fast R-CNN results

	Fast R-CNN	R-CNN
Train time (h)	**9.5**	84
- Speedup	**8.8x**	1x
Test time / image	**0.32s**	47.0s
Test speedup	**146x**	1x
mAP	**66.9%**	66.0%

Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

Source: R. Girshick